PUR-AM-11-012

PROJECT FINAL REPORT

“Acquisition Management for Systems-of-Systems: Risk Dynamics and Exploratory Model Experimentation”

Funded by the Naval Postgraduate School (NPS) Acquisition Research Program
Award No. N00244-10-1-0021

Submitted by:
Principal Investigator: *Daniel DeLaurentis*
Associate Professor, School of Aeronautics and Astronautics
Purdue University
701 W. Stadium Ave., West Lafayette IN 47907-2045
Ph : +1 765 494 0694 Fax: +1 765 494 0307
Email: ddelaure@purdue.edu

March 3, 2011

NPS Sponsor Contact:
Dr. Keith Snider
Associate Professor of Public Administration and Management
Acquisition Area Chair
Graduate School of Business & Public Policy
Naval Postgraduate School, Monterey, CA 93943
Office: (831) 656-3621 Email: ksnider@nps.edu

Executive Summary

The Department of Defense (DoD) has placed a growing emphasis on the pursuit of agile capabilities via net-centric operations. The breadth of technological advancements in communication and sensing has generated exciting opportunities for battlefield systems to exploit collaboration to multiple effects. In this setting, systems able to interoperate along several dimensions increase the efficiency of the overall system-of-systems (SoS) manifold. However, the manner in which these system-of-systems are acquired (designed, developed, tested and fielded) hasn't completely kept pace with the shift in operational doctrine. In our current project, we have attempted to unravel the layers of complexities in an SoS acquisition program, outline an acquisition strategy better suited for such programs and develop an exploratory analysis tool to provide insights into the acquisition process.

The research efforts during the report period have focused on the development of two types of tools to investigate the impact of development dependencies on the successful acquisition of SoS: 1) a computational exploratory model and 2) an analytical approach.

The conceptual model for acquisition strategy proposed in our project is based on the 16 technical management and technical system-engineering processes outlines in the Defense Acquisition Guidebook (DAG), often referred to as the 5000-series guide. Our conceptual model for acquisition is centered on the revised processes of the 2007 System-of-Systems System Engineering (SoS-SE). Simulation of the development process, however, is not always feasible due to the large amount of information required to perform a simulation. The analytical approach seeks to augment the computational approach by developing a method that enables the comparison of networks of systems that are connected to each by developmental interdependencies by exposing and quantifying the cascading effects of development risk. The goal is to allow acquisition professionals to develop intuition for procuring and deploying system-of-systems.

Applications of the modeling tool and the analytical network analysis method to different SoS topologies enables the comparison of the developmental performance of differing SoS and provides the ability for acquisition professionals to identify the features of a SoS that contribute most to the success or delays in the development process.

The work summarized in this report builds on the progress made during the previous year's work by improving upon and experimenting with the exploratory model to address systems-specific risk, its propagation to interdependent systems, and the comparison of SoS alternatives; this last item is possible by both the exploratory model as well as the analytical network analysis method also developed during this period. Example studies presented in this report have quantified the importance and impact of system-specific characteristics (e.g. development risk, development pace, interdependency strength, etc.) as well as their cascading effects for the entire SoS. The example studies present the potential of these tools to aid acquisition professionals to develop an intuition of the development of complex SoS and methods that enable the analysis of alternatives for SoS in the context of the development and acquisition process.

Outreach & Collaboration: Our work during the project period has resulted in *five external publications/presentations* at a diverse set of venues.

First and foremost was our paper and presentation at the 7th Annual NPS Acquisition Research Symposium in Monterey, CA in April 2010. This event continues to be a valuable venue for obtaining both critique of our work from fellow academics but also recommendations and interest from the practitioner community. Such broad-based feedback is unique to the Symposium. The citation for this paper is: *Mane, M., DeLaurentis, D.A., "System Development and Risk*

Propagation in Systems of Systems," 7th Naval Postgraduate School Symposium*, Monterey, CA, 11-13 May 2010.*

Second, an invited presentation on this work was made at the 2nd Annual Review of Research for the Systems Engineering Research Center (SERC). The SERC, funded by the DDR&E and NRO primarily, was interested in hearing new ideas in systems engineering research that was funded elsewhere but that could be leveraged in new ways for SERC specific research objectives.

Third, a paper that focused on the analytical portion of the work (described later in this report) was prepared and presented. The citation for this paper is: *Mane, M., DeLaurentis, D.A., "Network-Level Metric Measuring Delay Propagation in Networks of Interdependent Systems," 5th IEEE International Conference on System of Systems Engineering, Laughborough, UK, 22-24 June 2010*. The exposure to researches in system of systems modeling was a valuable experience for us; several suggestions have been incorporated to improve the approach.

Fourth, a paper was presented at an international aerospace engineering conference to broaden the exposure of the application problems that have served as our examples for demonstrating the methods we develop. The citation for this work is: *Mane, M., DeLaurentis, D.A., "System Integration and Risk Propagation in Aeronautical Systems-of-Systems," The 27th Congress of International Council of the Aeronautical Sciences (ICAS), Nice, France, 19-24 Sept. 2010*.

Finally, a journal article stemming from the analytical work was submitted to a special issue on complexity of the ASME Journal of Mechanical Design.

Introduction

The purpose of capabilities-based acquisition, as described by Charles and Turner (2004), is to acquire a set of capabilities instead of acquiring a family of threat-based, service-specific systems. The Missile Defense Agency (MDA), for example, uses capability-based acquisition to evaluate the success of a program based on its ability to provide a new capability for a given cost, and not on its ability to meet specific performance requirements (Spacy, 2004). The Joint Mission Capability Package (JMCP) concept is another example that aims to create a joint interdependency between systems to combine capabilities in order to maximize reinforcing effects and minimize vulnerabilities (Durkac, 2005). The goal is a more efficient utilization of both human and machine-based assets and, in turn, improved combat power.

To accomplish the desired capability, systems are increasingly required to interoperate along several dimensions which characterizes them as systems-of-systems (SoS) (Maier, 1998). Systems-of-systems most often consist of multiple, heterogeneous, distributed systems that can (and do) operate independently but can also collaborate in networks to achieve a goal. Examples of systems-of-systems include: civil air transportation (DeLaurentis et al., 2008), battlefield ISR (Butler, 2001), missile defense (Francis, 2007), etc. According to Maier (1998), the distinctive traits of operational and managerial independence are the keys to making the collaboration work. The network structure behind the collaboration, however, can contribute both negatively and positively to the successful achievement of SoS capabilities and, even earlier, to the developmental success. Collaboration via interdependence may increase capability potentials, but it also contains concealed risk in the development and acquisition phases. Brown and Flowe (2005), for instance, have investigated the implications of the development of SoS to understand the drivers that influence cost, schedule, and performance of SoS efforts. Results of their study indicate that the major drivers – as indicated by subject-matter-experts – include systems standards and requirements, funding, knowledge, skills and ability, system interdependencies, conflict management, information access, and environmental demands.

Disruptions in the development of one system can have unforeseen consequences on the development of others if the network dependencies are not accounted. The goal of a single system's program manager is the mitigation of risk leading to successful development of that specific system. While direct or immediate consequences of decisions are nearly always considered, the cascading second-and-third order effects that result from the complex interdependencies between constituent systems in a SoS are often not, which make success all the more difficult. It falls on acquisition managers and systems engineers (or systems-of-systems engineers) to understand and manage the successful development of a system, or family of systems, to produce the targeted capability in this challenging setting.

Evidence is abundant that system-of-systems oriented endeavors have struggled to succeed amidst the development complexity. The Future Combat System is a latest example (Gilmore, 2008). Civil programs have not been spared either, e.g. Constellation Program (Committee on Systems Integration for Project Constellation, 2004) and NextGen (NextGen Integration and Implementation Office, 2009). Rouse (2001) summarizes the complexity of a system (or model of a system) as related to the intentions with which one addresses the systems, the characteristics of the representation that appropriately accounts for the system's boundaries, architecture, interconnections and information flows, and the multiple representations of a system.

The work summarized here specifically targets complexities stemming from system development risk, the interdependencies among systems, and the span-of-control of the systems or system-of-systems managers and engineers. *The objective of the research is to quantify the impact* *of system-specific risk and system interdependency complexities using a) our evolving computational exploratory modeling approach and, b) a new analytical approach for quantifying the same effects.* The work comprises new improvements to a computational exploratory model (CEM) – a discrete event simulation model – previously introduced in prior Acquisition Symposia (Mane and DeLaurentis, 2009 and 2010) that aims to provide decision makers with insights into the development process by propagating development risk in the SoS network and capturing the impact that system risk, system interdependencies, and system characteristics have on the timely completion of a program. We also introduce complementary work related to an analytical approach to treat the same complexities via computations on conditional probabilities that relate the transmission of risk in network dependent systems.

Computational Exploratory Model (CEM) Overview

The CEM is based on the 16 basic technical management and technical system-engineering processes outlined in the Defense Acquisition Guidebook (U.S. Department of Defense, 2008a), often referred to as the 5000-series guide. However, an SoS environment changes the way these processes are applied. The Systems Engineering Guide for System-of-Systems (SoS-SE) (U.S. Department of Defense, 2008b) addresses these considerations by modifying some of the 16 processes in accord with an SoS environment. The resulting processes and respective functions consist of translating inputs from relevant stakeholders into technical requirements, developing relationships between requirements, designing and building solutions to address requirements, integrating systems into a high-level system element, and performing various managing and control activities to ensure that requirements are effectively met, risks are mitigated, and capabilities achieved.

The CEM, centered on these revised processes, is a discrete event simulation of the development and acquisition process. This process creates a hierarchy of analysis levels: SoS Level (L1), Requirement Level (L2), and System Level (L3). Component elements at each level are a network representation of the level below. The SoS Level (L1) is comprised of the numerous, possibly interdependent, requirements (L2) needed to achieve a desired capability. Similarly, satisfaction of each requirement in the Requirement Level (L2) requires a number of possibly interdependent systems (L3). Figure 1 presents the description of the process modeled by the CEM.

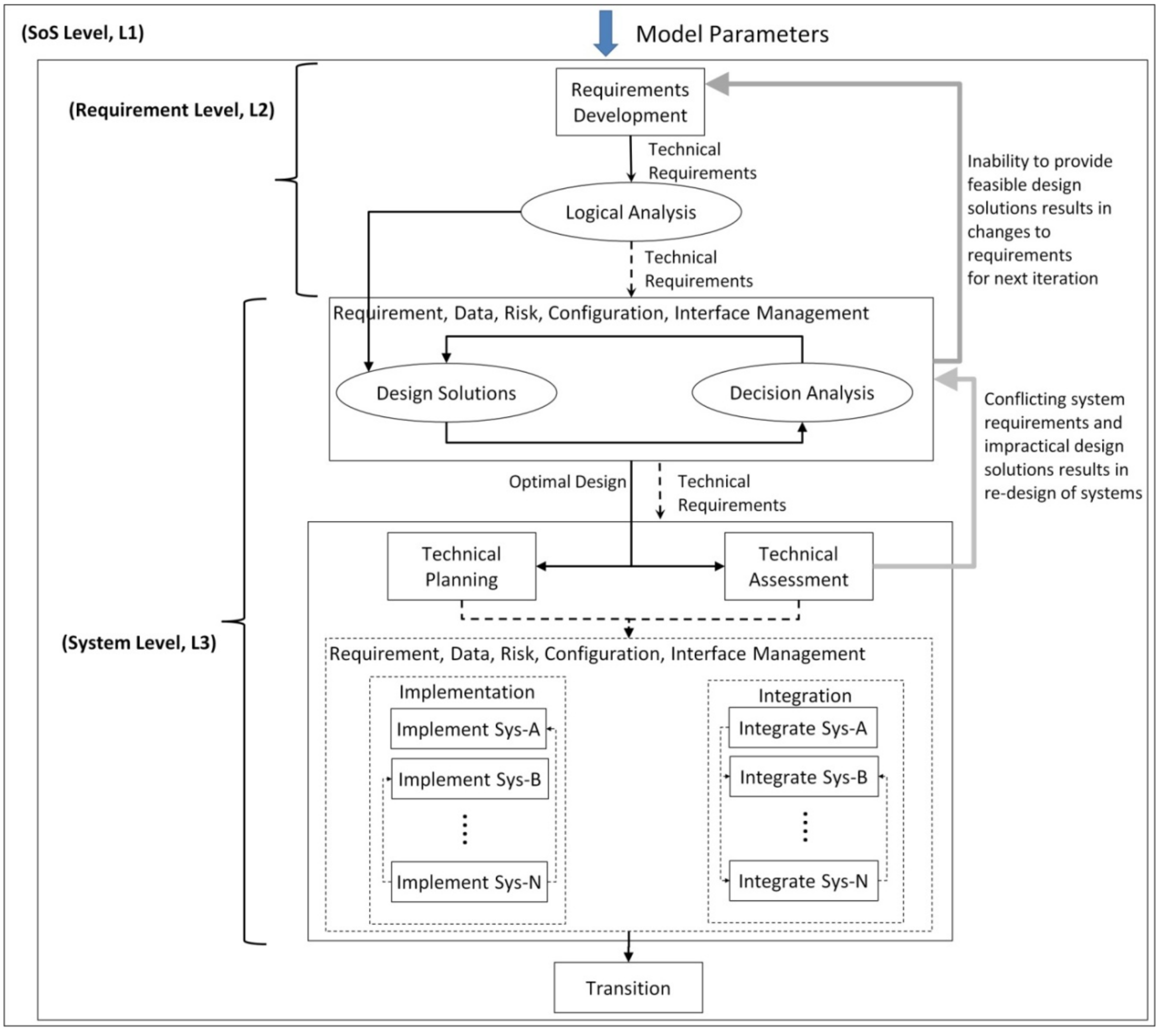


Figure 1. **Conceptual model of Acquisition Strategy based on SoSE Process (described in U.S. Department of Defense, 2008b)**

At the Requirement Level (L2), *Requirements Development* contains the technical requirements of the SoS (provided externally). The technical requirements are then examined in *Logical Analysis* to check for interdependencies amongst the requirements. A check for inconsistencies amongst requirements is also performed. *Design Solution* development and *Decision Analysis* are the next processes, which belong to the System Level (L3). They produce the optimal design solution from the set of feasible solutions to meet the given requirements. The optimal design solution is based not only on the current set of requirements and solution alternatives but also takes into account all previous information available through requirements, risk, configuration, interface and data management processes. Because most acquisitions are multi-year projects involving many different parties, the overlap between the management processes, *Design Solution* and *Decision Analysis* processes, allows for greater tractability of decisions. It is at this stage that system interdependencies are identified. The optimal design solution obtained from this phase is then sent to the next stage: *Technology Planning* and *Technology Assessment*. In the event that an optimal or sub-optimal design solution to successfully implement the given requirements does not exist, the feedback loop to *Requirement Development* translates into a change in the technical requirements for the SoS. *Technology Planning* and

Technology Assessment are System Level (L3) scheduling processes that oversee the implementation, integration, verification and validation for all the component systems in the SoS.

Systems in the SoS are often dependent on other systems for either implementation, integration, or both. Disruptions during these stages of development in one of the systems result in time-lags in the acquisition process and to delays that propagate through the network of component systems impacting seemingly independent systems. For example, if the implementation of a system A is dependent on the Implementation of a system B – as could be the case for the development of an aircraft that depends on the specifications of a radar system – funding cuts to system B can result in development delays in system B but can also impact the development of system A. If, on the other hand, a third system C depends on system A, this could also be affected by the problems caused in system C due to funding cuts.

The *Implementation* and *Integration* Phases of component systems constitute the lowest level of detail modeled in the CEM. The design decisions made at earlier stages must be implemented and integrated in these phases to generate the final product of a program. Figure 2 presents an abstraction of the layered networks that result from the modeling of the acquisition process: systems are grouped to satisfy a requirement, and requirements are grouped to generate a capability.

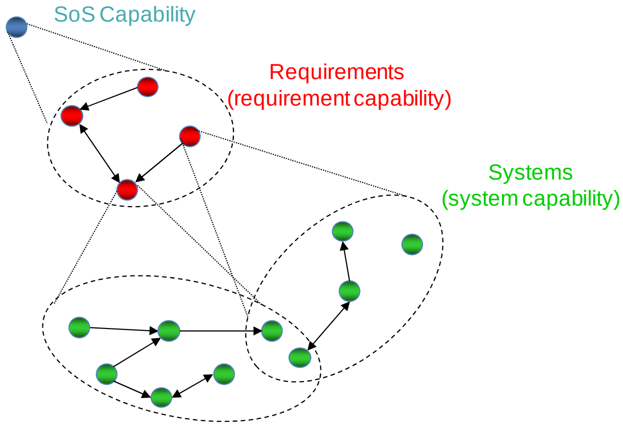


Figure 2. Layered network abstraction of Computational Exploratory Model

Systems can be independent, can satisfy several requirements, and can depend on other systems. The CEM simulates these layered relationships to capture the impacts that any changes – related to decision-making, policy, or development – in any of the component systems, requirements, and relationships between them have on the completion of a project. The exercise of the CEM described in this paper specifically targets complexities stemming from system risk, the interdependencies among systems, and the span-of-control of the SoS authority (if present). The next section will present the model dynamics that make possible the study of these complexities and will explore the design space of the SoS authority and tradeoffs between development risk and the number of systems and system interdependencies in a SoS.

Detailed Model Dynamics

The CEM operates as a discrete event simulator of the development process. Several challenges arise in developing a model for purposes of simulation and learning. Disruptions occur at various stages of development and are governed by the risk associated with the project or individual systems. The CEM models risk associated with the implementation and integration of each component system as well as the risk due to the system interdependencies. Furthermore, systems and SoS engineers are often faced with the decision of using legacy assets to satisfy a given requirement or opt for the development of brand new ones. The CEM includes parameters such as *readiness-level* to differentiate between legacy assets/platforms, new systems, and partially implemented/integrated systems (i.e. systems under development) and to investigate the impact that the inclusion of such systems in the development of an SoS has on the success of a project. The next sub-sections describe the model details: parameters and inputs, Implementation and Integration dynamics, and the risk model.

Model Input Parameters

Table 1 presents the input parameters and the remainder of this section expands and explains their role in the CEM.

Table 1. Input parameters of computational exploratory model

Parameter	Notation	Description
		Requirement Level (L2)
Requirement dependencies	D_{req}	Adjacency matrix that indicates requirement interdependencies
Risk profile	R_{req}	Probability of disruptions in *Requirement Development Phase*
Impact of disruptions	I_{req}	Time penalty when disruptions hit *Requirement Development Phase*
		System Level (L3)
System dependencies	D_{sys}	Adjacency matrix that indicates system interdependencies
Development pace of design	t_{des}	Increase in completion of *Design Solutions Phase*
Design risk profile	R_{des}	Probability of disruptions in *Design Solutions Phase*
Impact of design disruptions	I_{des}	Time penalty when disruptions hit *Design Solutions Phase*
Span-of-control	*soc*	Indicator of how *Implementation* and *Integration* are performed (sequentially or simultaneously)
System initial readiness-level	$m^0(i,r)$	Initial readiness-level of system *i* to satisfy requirement *r* (for *Implementation Phase*)
System risk profile	$R_{sys}(i,r)$	Probability of disruptions (during implementation) of system *i* when satisfying requirement *r*
Impact of disruptions	$I_{sys}(i)$	Time penalty when disruptions hit system *i* during Implementation/Integration
Implementation pace	$p_{imp}(i)$	Increase in readiness-level at each time step during implementation of system *i*
Integration pace	$p_{int}(i)$	Increase in completeness-level at each time step during integration of system *i*
Implementation start	$l_{imp}(i,j)$	Readiness-level of system *j* when *Implementation Phase* of dependent system *i* begins
Strength of dependency	$S(i,j)$	Strength of dependency of system *i* on system *j*

The requirement dependency matrix (D_{req}) indicates how the development and satisfaction of requirements depend on each other, which impacts the sequence in which requirements are developed and satisfied. For example, if Requirement A depends on Requirement B, then development of Requirement A begins when Requirement B has been satisfied. As requirements are developed, the risk profile (R_{req}) of *Requirement Development* indicates the probability of disruptions at this stage in the development process. Disruptors signify a change in requirements or addition of new requirements. When a requirement is changed after the acquisition process has begun, it affects all subsequent processes and it causes a time delay (I_{req}) that is added to the project time. Every requirement that is implemented is fed into its own *Design Solution* and *Decision Analysis* (Figure 1) process. The *Design Solution* and *Decision Analysis* processes feed into each other and the risk profile (R_{des}) indicates the probability of disruptions at each time–step during the completion of the stage with a value between 0 and 1. Any disruptions at this stage indicate that the design solution provided is not feasible and a time penalty (I_{des}) that indicates a re-design of the solution is incurred. If the solution fails in multiple consecutive time-steps, then the requirement is sent back to *Requirement Development* stage, otherwise the set of component systems and their user-defined parameters are sent to the *Technical Planning* and *Technical Assessment* (Figure 1) processes based on the development-pace parameter of this stage.

Implementation Phase Dynamics

Technical Planning is the stage where *Implementation* and *Integration* of component systems is performed. The *Implementation Phase* simulates the development of each system. The nature of candidate systems may range from legacy systems to off-the-shelf, plug-and-play products to custom-built, new systems. Development of a 'brand new' SoS has been and will remain a rare occurrence. In their study on SoS, the United States Air Force (USAF) Scientific Advisory Board (Saunders, 2005) stated that one of the challenges in building an SoS is accounting for contributions and constraints of legacy assets. Similarly, the regular utilization of off-the-shelf component systems in both defense and civil programs contribute to cost and time savings but also introduce a different type of risk to the system development process (Constantine, 2010). These legacy systems may be used 'as-is' or may need re-engineering to fulfill needs of the new program.

Here, we define *legacy systems* as systems that have been developed in the past to achieve a particular requirement, and *new systems* as not-yet-developed systems envisioned to satisfy a new requirement. When considering the use of legacy systems to meet a new requirement, the capability of these systems to satisfy the new requirement is not necessarily the same as their capability to meet the original requirement for which they were designed. Additionally, the risk associated with the modification of a legacy system and the risk associated with the development of a brand new system can be quite different. Legacy systems may, however, provide cost and/or time benefits if modifications are less severe than a new development, as is the case with new systems. To delineate systems in a meaningful way, we describe the spectrum of a system's ability to satisfy a requirement in terms of its readiness-level.

System readiness-level, a concept proposed by Sauser et al. (2006), is a metric that incorporates the maturity levels of critical components and their readiness for integration (i.e. integration requirements of technologies). This is an extension of the widely used Technology Readiness Level (TRL), a metric that assesses the maturity level of a program's technologies before system development begins (Department of Defense Directive 5000.2, 2005). While similar in spirit to the SRL metric proposed by Sauser et al. (2006), readiness-level in the present work is defined in a different manner and with less detail. We define system readiness level as the readiness-level of a system i to satisfy requirement r, $m(i,r)$, with a value between 0 and 1. A system with a readiness-level of 1 is a fully developed system that can provide a certain level of capability. The dynamic model starts the *Implementation Phase* of a system from its initial readiness-level and simulates its development / implementation until it reaches a readiness-level of 1. An initial readiness-level of 0 indicates a brand new system that must be developed from scratch, while a system with an initial readiness-level greater than 0 indicates a legacy system that is partially developed to satisfy a requirement r, but needs further development to reach a readiness-level of 1. In general, careful research of a candidate system i will determine its initial readiness-level to satisfy a requirement r, and, therefore, the amount of development necessary to achieve a readiness-level of 1.0.

The CEM simulates the *Implementation Phase* as a series of time steps in which a pre-determined increment of readiness ($p_{imp}(i)$) is gained at each time-step of each system i, or lost if a disruption occurs (according to the system risk profile of system i in satisfying requirement r,

$R_{sys}(i,r)$). This is clearly a gross simplification of the actual development process for a system; however, it adequately serves the purposes of the research, which is focused on the interdependencies between systems to develop a SoS capability and aims to capture the impact of disruptions on the development process. Accurate modeling of the *Implementation Phase* would increase the accuracy of the model for a particular application but it would not change the nature of the observed results.

Representation of Risk

The risk associated with the development of a system is a function of its inherent characteristics (technology, funding, and complexity levels) and on risk levels of the systems on which it depends. The former may be estimated via a variety of analysis techniques that examine a system in detail, but the latter requires knowledge of system interdependencies which can be numerous, complicated, and often opaque. Developmental interdependencies of SoS create layered networks that often span among a hierarchy of levels (DeLaurentis et al. 2005, Butler et al. 2001, Ayyalasomayajula et al. 2008, Kotegawa et al. 2008). The complexity of these networks often hides many of the otherwise explicit consequences of risk. Depending on the network topology characteristics, disruptions to one of the critical nodes or links in the network can propagate through the network and result in degradation to seemingly distant nodes (Huang et al. 2008).

In this study we express risk as a density function that describes the probability of a disruption occurring at any time during the system development. We concentrate on the *Implementation* and *Integration Phase* as the development stage where disruptions occur. Here, inherent risk is the probability of disruptions due to the development characteristics of the subject system, e. g. technology readiness-level, funding, politics, etc. Risk due to interdependencies, on the other hand, is the probability of disruptions during the *Implementation Phase* of a system due to disruption in the system on which the system of interest depends. This is essentially the conditional probability of a disruption given that another system has a disruption.

This study assumes that the inherent risk of a system *i* in satisfying requirement *r*, $R_{sys}(i,r)$, is solely a function of its readiness-level, $m(i,r)$. While a somewhat simplified definition, expressing risk as a function of a system's readiness-level is logical. Recall that readiness-level is a metric that describes the necessary development of a system to satisfy a given requirement. Therefore, risk changes as the readiness-level of a system increases. The following equation introduces a relationship between a system's readiness-level and risk (probability of disruption).

$$R_{sys}(i,r) = \alpha_i\left(1 - m(i,r)^{\beta_i}\right) \tag{1}$$

In this relationship, α_i (with a value between 0 and 1) is parameter that indicates the upper bound value of risk for system *i* (i.e. producing maximum probability of disruption) while β_i is a shape parameter that indicates how quickly risk changes as a function of readiness-level. This formulation implies that risk is highest at the early stages of development (e.g. low readiness-levels) and it decreases (at different rates depending on the value of the β_i parameter) as development progresses. For instance, when a system *i* has a readiness-level of 0.0 – it is a brand new system – the probability of disruptions during development will be highest, and it will have a value α_i. However, when the system has a readiness-level of 1.0, the probability of disruptions will be 0. System inherent-risk is implemented in the CEM by using a uniform random distribution to select a value between 0 and 1 at each time-step of the *Implementation* or *Integration Phase* and passing it into a binary channel to see if the number is smaller or greater than the probability of disruption defined by $R_{sys}(i,j)$. This determines if a disruption occurs or not.

When all systems are independent, identification of the system with highest risk is trivial (e.g. system that, on average, will contribute more to delays in completion time). However, when systems are interdependent, systems that otherwise have a low inherent risk can be greatly impacted by disturbances because of the transmission of risk from other systems. Systems are impacted by nearest neighbors (those systems on which they directly depend; first-order dependencies) and by systems that impact those nearest neighbors (higher-order dependencies).

The CEM models risk due to interdependencies in terms of the dependency strength between two given systems. Dependency strength, $S(i,j)$, is an input parameter that takes values between 0 and 1 and is defined as the conditional probability (uniform random probability) that system i has a disruption given that system j (on which system i depends) has a disruption. Risk due to interdependencies is, therefore, a function of the readiness-level of the dependent-upon system as well as the strength of that dependency. A notional example of a simple SoS is utilized here to present these features of the CEM (Figure 3).

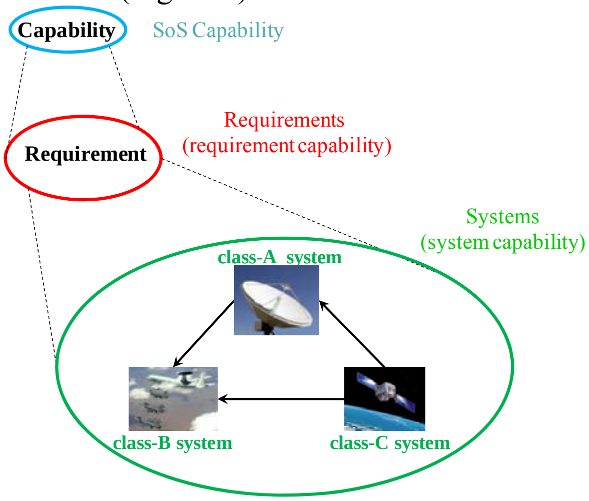
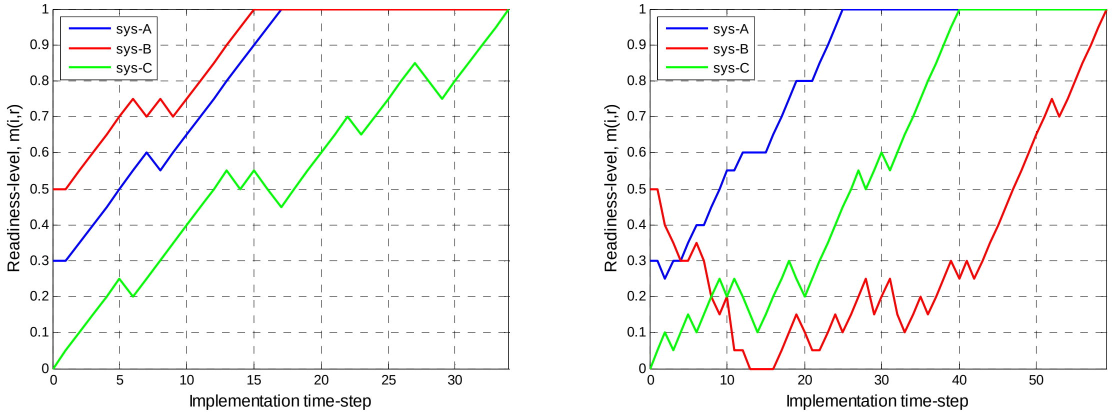


Figure 3. Layered network structure of example SoS

Each system in this simple SoS network serves a role and provides a certain level of capability in order to satisfy some requirement. The links between systems indicate interdependencies among systems. The arrows indicate the directionality of dependence, including the case of mutual dependence. Mane and DeLaurentis (2009) contains more detailed information on the CEM structure. For this example, Figure 4 presents the implementation history of this three-system SoS with a risk profile that has α_i, and β_i, values of 0.2 and 4, respectively, and two different levels of interdependency strength, $S(i,j)$.



a) $S(i,j) = 0$ b) $S(i,j) = 1$

Figure 4. Implementation Phase history for example problem

Each system has a different initial readiness-level – system-A of 0.3, system-B of 0.5, and system-C of 0. Recall that an initial readiness-level greater than zero indicates a legacy system that must be further developed to achieve a readiness level of 1 to satisfy a given requirement. The model assumes that the readiness-level of a system can reduce to below initial readiness-level value. This is reasonable since inherent disruptions or disruptions due to interdependencies can result in modifications to subsystems that were not previously considered (i.e. unforeseen technology limitations of a system may require redesign of a dependent system). In Figure 4a, all systems are independent (dependency strength of zero). The occasional set-backs in the readiness-level of each system are due to disruptions stemming from the inherent system risk. In Figure 4b, one the other hand, dependency strength is highest (with a value of one). Recall that dependency strength indicates the probability of disruption on the dependent system given that the system on which it depends has a disruption. When the dependency strength is one, a disruption in a given system is always propagated to the dependent systems. For example, disruptions in the development of system-C propagate to system-A with probability 1 and disruptions in the development of system-A propagate to system-B with probability 1. Note, for instance, that there is a reduction in readiness-level in the development of the system-B every time that there is a reduction in readiness-level during the development of system-A or system-C (on which system-B depends). The candidate systems for a desired capability can, in general, have different levels of dependency strengths.

The number of Developmental Information Elements shared between constituent systems of an SoS may vary with each system pair. In a given SoS, a value of dependency strength equal to 1 will be assigned to the system pair that shares the maximum amount of Developmental Information Elements. All other dependency strength values within the SoS will be relative to this maximum value. **Table 2** presents dependency strength associations, notional descriptions and associated values.

Table 2. Strength of Dependency of system *i* on system *j*, *S*(*i,j*) when the maximum number of Developmental Information Elements shared within any two constituent systems of the SoS is four.

Value, *S*(*i,j*)	Notional Definition
1.00	System *i* depends on system *j* for all 4 developmental information elements
0.75	System *i* depends on system *j* for 3 of the 4 developmental information elements
0.50	System *i* depends on system *j* for 2 of the 4 developmental information elements
0.25	System *i* depends on system *j* for 1 of the 4 developmental information elements
0.00	System *i* is completely independent of system *j*

Different dependency strengths and different inherent probability of disruption profiles may lead to different conclusions. For instance, though inherent probability of disruption may be high, impacts on total completion time may be small if the dependency strength is low. Figure 5 presents a sensitivity of development time for this example problem on the value of dependency strength.

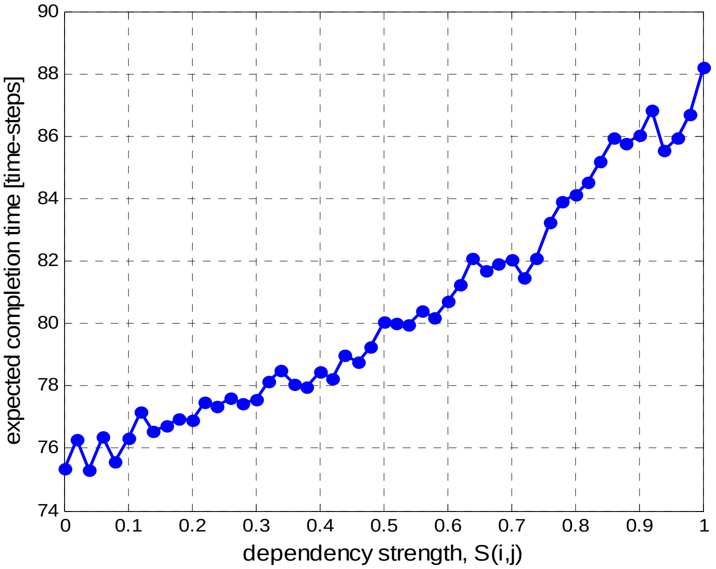


Figure 5. Impact of dependency strength on completion time for example problem

As expected, higher dependency strength means higher development time. In this example, the number of systems and interdependencies is invariable, and the increase in development time can be different for a different family of constituent systems. When considering the development of different families of systems that can provide a desired capability, the characteristics of interdependencies between component systems can have a large impact on the decision to pursue development of a certain alternative. Quantifying the impact that such characteristics have on the development process can aid decision makers in selecting the most promising alternative.

Impact of Risk and System Interdependencies

Quantifying risk is a complicated function of the individual system characteristics as well as the interdependencies between systems. The combinations of systems that can achieve a given capability-level can be numerous. Depending on the selection of the constituent systems, the completion time of a project can vary greatly due to the number of constituent systems, their interdependencies, and risk profiles. As these families of systems get larger, it becomes more difficult to quantify the impact that each system and system-characteristic has on the success of a project. For instance, a three-system solution may appear to be preferable to a ten-system solution; but the interactions between the three systems can result in disruption propagation that greatly impacts the timely completion of the project. System interdependencies and their characteristics can impact the completion time of a project by affecting the way in which disruption propagate. In this section we demonstrate the impact that system-inherent risk and the strength of interdependencies between component systems can have on the timely completion of a project. Furthermore, we show and quantify how different families of systems that can provide the same set of capabilities can have greatly differing development histories.

Interdependency Strength and Inherent Risk

For this investigation we assume that in order to achieve some capability a family of three classes of systems has been identified; for instance, a class-A system can be a land-based radar or an airborne radar; a class-B system can be a large transport aircraft, a mid-size aircraft, or a small aircraft. Each of these classes of systems provides a certain capability that is required to achieve a global capability of the SoS. The design authority must decide which constituent system to select for each system-class. A notional example of a simple SoS is utilized here (Figure 6).

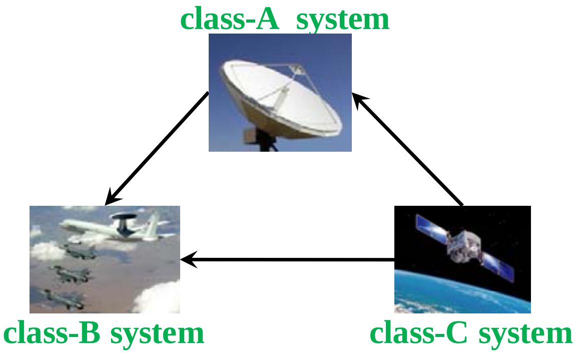
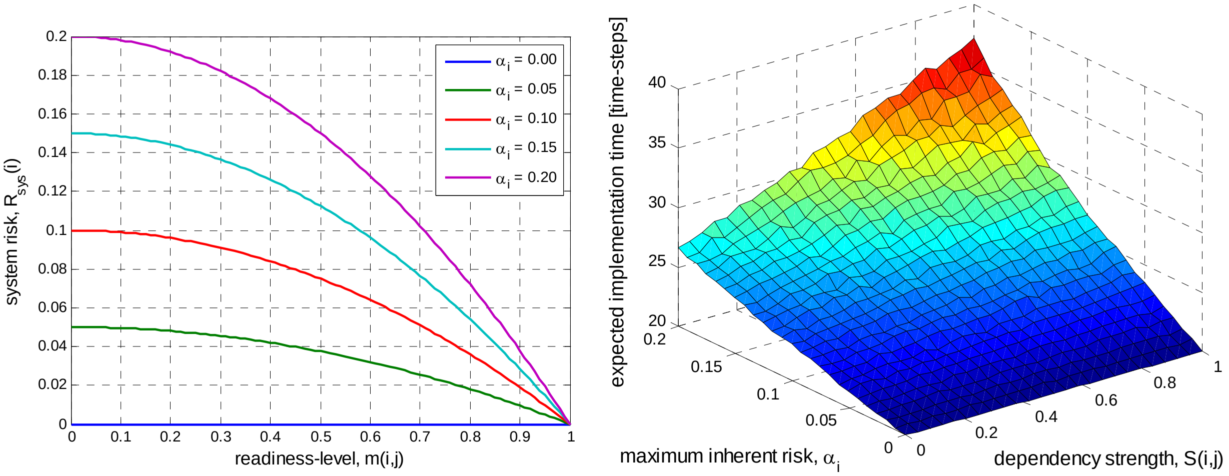


Figure 6. Interdependencies of notional SoS

The links between systems indicate interdependencies among systems. For instance, development of a class-B system must rely on information about the development and capabilities of a class-A system in order to continue development. Similarly, development of a class-A system needs information from a class-C system. Different systems are available to designers or systems engineers for each system-class. Each candidate system can have different risk characteristics as well as different interdependency characteristics. If we assume that the systems engineer has identified these characteristics for each candidate system, we can use the CEM to simulate the development process when different combinations of these candidate systems are considered and identify the family of systems that results in the lowest expected completion time. The strength of the CEM is in its ability to aggregate the individual system characteristics and quantify the SoS-level performance (with respect to development time) of a family of candidate systems.

Figure 7 presents results where the expected implementation time of a family of candidate systems is measured against the inherent risk of individual systems and their interdependency strengths.



a) System-specific risk profile b) Expected implementation time

Figure 7. Impact of risk due to interdependencies on implementation time

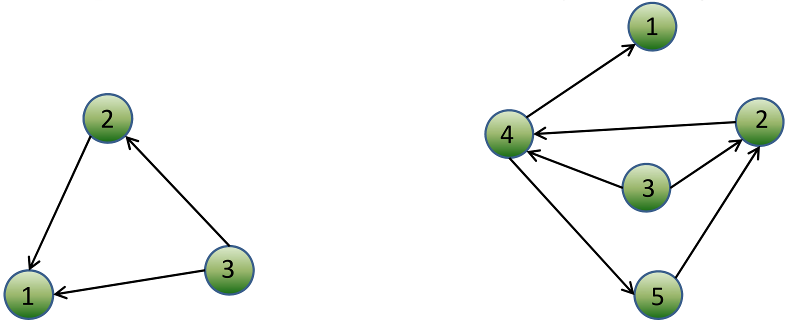
We assume here that all candidate systems will have the same risk profile and all interdependencies will have the same strength. Figure 7a shows the inherent system risk, $R_{sys}(i,r)$, as a function of system readiness-level, $m(i,r)$, for five different risk profiles (five different α_i values and a fixed β_i parameter of 2). The value of α_i indicates the maximum inherent risk of a system according to Eqn. 1. The assumption here is that risk is highest in the earlier stages of development and it decreases as development progresses. The results in Figure 7b present the expected implementation time when families of systems with different combinations of inherent risk profile and dependency strengths are considered. Each point on the surface indicates a family of candidate systems with a given combination of maximum inherent risk and dependency strengths. For instance, a solution that entails systems with a maximum inherent risk of zero and dependency strength of zero (e.g. independent systems with no development risk) will have an expected implementation time of 20 time units. The three systems are developed simultaneously but have no impact on each other's development. The trends in Figure 7b show that the impact on implementation time of families of systems that have strong interdependencies is larger than when the systems have high inherent risk but low dependency strengths (e.g. the increase in implementation time is smaller as inherent risk increases than when the strength of dependencies increases).

This investigation quantifies the impact that system interdependencies have on the implementation time of a project. The results presented here point out the importance of interdependencies in the development process. This type of analysis can prove useful to an SoS authority when selecting potential component systems as a part of a family of systems or SoS to satisfy a given requirement and achieve a desired capability.

This simple example considers families of systems comprised of three constituent systems. Different candidate families of systems, however, can have differing number of constituent systems that can provide different system-capabilities to achieve the desired SoS capability. Similarly, risk profile and interdependency characteristics of the constituent systems can result in different disruption propagation and different development solutions.

Comparison of Alternatives

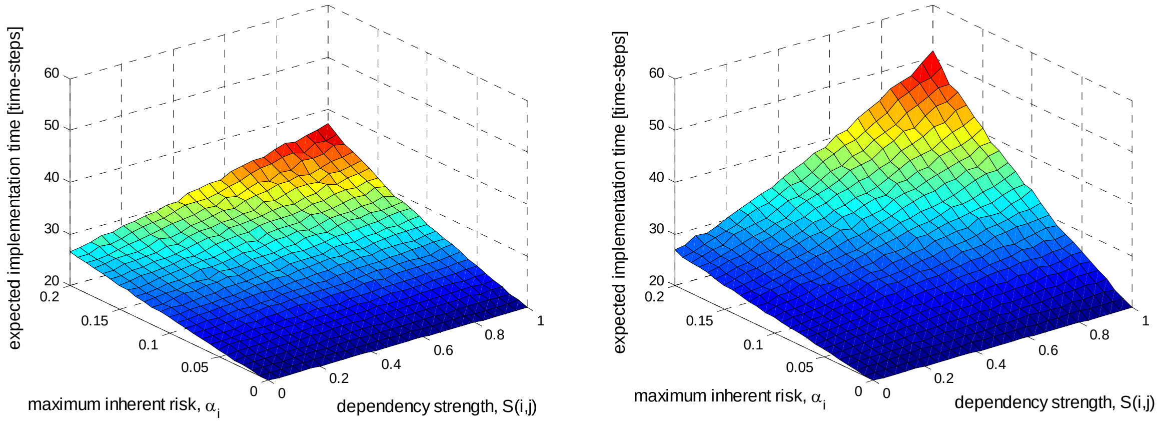
Given a set of alternative means to satisfy a requirement, an SoS authority (in conjunction with systems engineers) must determine the best network of systems to develop and acquire. The number of systems alone may not be a good indicator of the complexity of a system and the eventual developmental success. The risk profile of systems as well as the number and strength of system interdependencies play an important role that often hamper understanding of the impact of decisions. For instance, a SoS that is comprised of three constituent systems may appear more likely to succeed than a SoS comprised of five systems. However, the number and strength of interdependencies between the five systems may be such that the expected completion time of this SoS is lower than the expected completion time of the three-system SoS. The three-system example in the previous section showed that the strength of dependencies plays an important role in the timely completion of a SoS project. Here we use the CEM to investigate the impact that network characteristics (number of systems, number of dependencies, and strength of dependencies) have on the completion time of a SoS project. We compare the developmental time of two example-SoS comprised of three and five constituent systems (Figure 8).



a) Three-system alternative b) Five-system alternative

Figure 8. Alternative families of systems

The three-system network is the same network with three interdependencies as the one presented in Figure 6. The new, five-system network is clearly a larger SoS with more systems and six interdependencies. As in the previous section, different candidate systems are available to provide the required capability level. The systems engineer would like to quantify the expected implementation time of each combination of systems for the three-system and the five-systems options. Via a Monte Carlo simulation of 500 samples we are able to compute the expected implementation time of the five-system network – we previously did the same for the three-system network. Figure 9 presents this result for the different combinations of inherent system risk and dependency strengths.



a) Three-system alternative b) Five-system alternative

Figure 9. Expected implementation time of alternatives

Figure 9a presents the expected implementation time of the three-system option (the same as Figure 7b) while Figure 9b presents the expected implementation time of the five-system network. As in the previous analysis, these results indicate the expected implementation time of candidate component systems that have differing levels of inherent risk and interdependency strengths. The trends in the expected implementation time of the five-system option are larger than those of the three-system option. This is expected because the former has more systems as well as more interdependencies. Recall, however, that each point in these charts represents a candidate family of systems and one can see that the expected implementation times of some five-system alternatives are lower than some three-system alternatives. To show this more clearly Figure 10 presents the expected completion times when the inherent system risk of all candidate systems is highest ($\alpha = 0.2$).

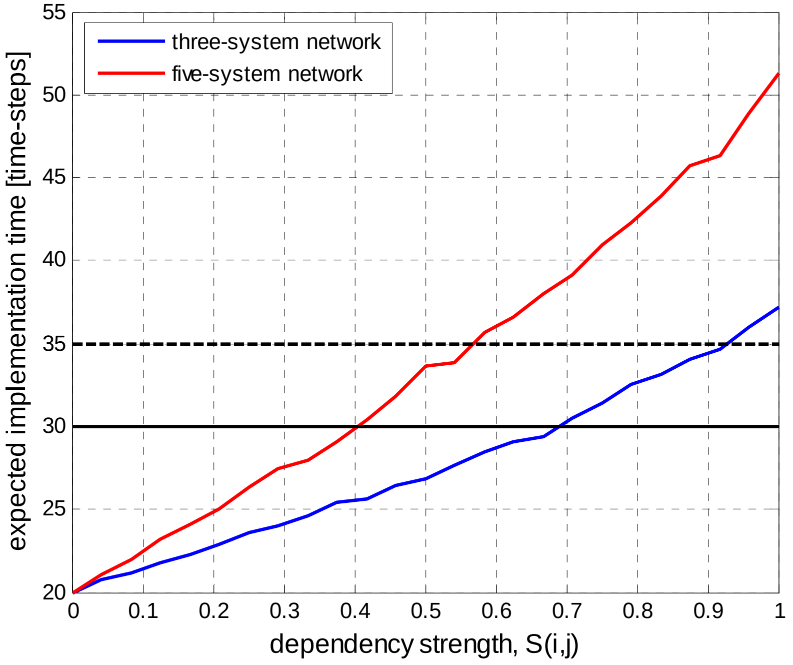


Figure 10. Expected implementation time of sample results

As previously mentioned, the implementation time of the five-system alternatives is always higher than the three-system alternatives. However, if the dependency strength between the systems in the three-system alternative has a value of 1, the expected implementation time of this alternative will be 37 time units; if the dependency strength between the systems in the five-system is not as strong, say with a value of 0.4, the expected implementation time will be 30 time units. Therefore, depending on the strength of the interdependencies between the constituent systems, a family of systems can be a better (lower expected implementation time) alternative. By simulating the development process of different alternatives via the CEM, it is possible to quantify the impact of system specific risk, the risk due to interdependencies, and the propagation of disruptions to compare different alternative solutions that can provide a desired level of capability.

Analytical Approach

Additional complexity in the model, carefully selected, will likely increase the efficacy of the CEM. However, as a simulation-based approach, it too has limitations. Therefore, in conjunction with the further development of the CEM, the investigators are also developing an analytical approach that captures the characteristics of a network that results from the developmental interdependencies of systems. This is an approach that uses a network-level metric to treat the same complexities via computations on conditional probabilities that relate the transmission of risk in networks of interdependent systems. This provides means to compare networks in their ability to arrest the propagation of delays caused by random disturbances and can be used as a figure of merit when designing SoS architectures that aim to achieve some desired capability.

While typical networks like the World Wide Web, social networks, and communication networks are a result of evolution, the networks created by the development of interdependent systems can be designed. Being able to quantify the performance of such networks enables comparison of networks, and ultimately the design of networks that optimize that performance. During development, the ability of a network of systems to propagate or arrest disruptions can be an important performance parameter when selecting a family of systems to provide a certain level of capability.

Network analysis tools can help to describe the properties of a network and to identify critical component systems. The number of links and nodes in a network, for instance, can indicate the complexity of a network by measuring the number of systems and their link. Similarly, network average degree, which describes the average number of links of each node, can indicate the level of connectivity in a network and help identify critical systems. These traditional network measures, however, are unable to describe the performance of the entire network and, consequently, comparison of networks in their ability to arrest the propagation of disruptions that can create development delays.

Delay propagation modeling is common in the airline industry, where delays at one airport can easily propagate in the aviation network and impact dependent airports. Approaches for modeling and estimating these delays, however, center on regression analysis [Xu et al. (2005) and AhmadBeygi et al. (2008)]. AhmadBeygi et al. (2005), for instance, investigates the relationship between the potential propagation of flight delays to subsequent flights and the utilization levels of air service providers. Even though the delays can propagate indefinitely, the delay propagation structure is acyclic. A delay caused by mechanical problems to an aircraft (flight number) will always propagate forward. In the system development process, delays can be cyclic, which increases the complexity of the problem and limits the ability of current approaches to quantify the total delay.

This research presents a network-level metric that captures the characteristics of a network that results from the development interdependencies of systems and it provides means to compare networks in their ability to arrest the propagation of delays caused by random disturbances.

Delay Propagation

SoS systems acquisition involves the development of many interdependent systems. Interdependencies impose constraints in the development of individual systems and couple their development. Of concern is the propagation of delays that result from random disruptions in the development process of a particular system. Disruptions in the development of one system can impact the development of a dependent system. For instance, a problem in the development of one system can mean that systems that depend on it for information may experience delays, which can further delay the systems that depend on these systems. Depending on the strength of interdependencies and/or the likelihood of a disruption to propagate, the network-level impact of a system-specific disruption can be much larger than its impact on the original system that experienced the disruption. Quantifying the network-level impact of disruptions as a function of network characteristics can be a powerful means to compare networks.

The approach proposed here to measure the performance of networks in their ability to arrest the propagation of delays is based on the classical "lost miner problem" (Ross, 2007). In this example problem, a miner is lost in a cave inside a mine where there are four tunnels that lead out of the cave, but only one leads out of the mine (Figure 11).

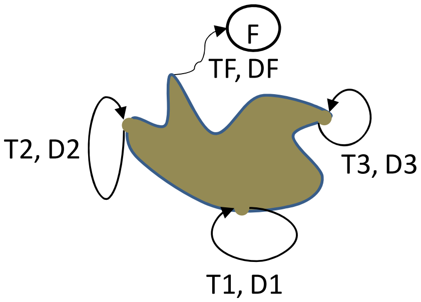


Figure 11. Lost-miner problem

The miner can choose to enter a tunnel T_i with probability $P(T_i)$ and has no memory of his previous choice. If the miner chooses tunnel T_1 he wanders in the tunnel for D_1 days and returns to the cave, where he must decide which tunnel to enter next. If he chooses tunnel T_2 or T_3 he wanders in the tunnel for D_2 or D_3 days, respectively, and returns to the cave. If he chooses tunnel T_F, he is free, instantly. The question the problem poses is: What is the expected time until the miner reaches freedom (e.g. the expected duration of the miner's stay in the mine)?
We can describe the delay propagation in the system development process following the same reasoning. We describe a network of systems in terms of the number of systems (caves), the number and direction of their dependencies (tunnels), and the characteristics of the interdependencies (probability of choosing a given tunnel), e.g. probability of passing-on a disruption and the impact of the disruption. The simple three-system network below is used to describe the proposed approach.

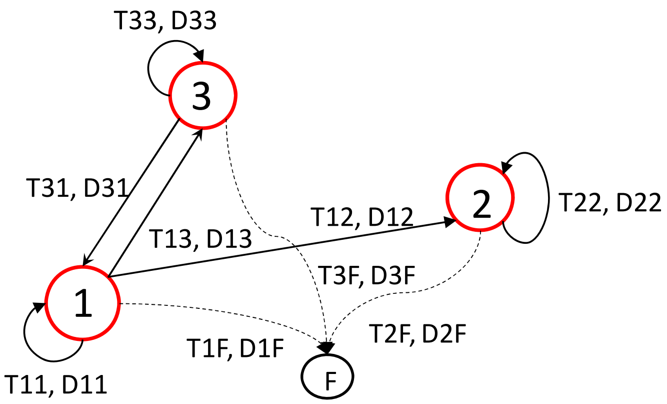


Figure 12. Example systems development network

Each node represents a system that is under development (i.e. aircraft, missile, radio) to achieve some capability. The links indicate interdependencies between the systems as well as the strength of those interdependencies. For instance, system-1 depends on system-3 because information from system-3 is needed to continue development of system-1. T_{ij} represents the probability that a disruption in the development of system *i* will impact development of system *j* and D_{ij} represents the impact of a disruption (delay) on system *i* that propagates to system *j*. These two quantities represent the strength of the dependency between system *i* and system *j*. Two systems can be strongly dependent if the probability of a disruption propagating from one system to the other is high or if the delay experienced by one system because of a disruption in the development of the other is large. Node F is a sink that represent the arrest of the propagation of delay in the network. In this setting, a disruption can be seen as an event that travels from system to system causing development delays until it exits the network (via node F). This is similar to the "lost miner problem" where the miner chooses tunnels until he reaches freedom.

In system development disruptions can be a result of funding decisions, political environment, technological setbacks, etc. For example, system-1 can be faced with budget cuts and the program manager must reduce funding to one of the subsystems that comprise system-1. Depending on the magnitude of the reduction in funding, this can have no impact in the development of system-1 with probability T_{1F}, and nothing is affected; it can cause a delay of D_{11} days with probability T_{11} in the development of system-1 that is not sufficiently large to impact dependent systems; or it can result in a delay of D_{13} days with probability T_{13} that impacts development of system-3. Additionally, the delay in the development of system-3 can cause further problems that delay its development by D_{33} days with probability T_{33}; it can cause a delay of D_{13} days with probability T_{13} that creates a problem in the development of system-3; or a delay of D_{31} days with probability T_{31} that impacts system-1; or, conversely, the problem is not sufficiently large to cause any delays with probability T_{3F}, and the propagation of delay in the network is arrested.

Depending on the strength of the dependencies between systems and the magnitude of disruptions, delays can propagate and accumulate in a network. Hence, networks with different number of systems, interdependencies, and strength of interdependencies will perform differently when faced with random disruptions. The ability to estimate the expected accumulation of delays as a function of these network characteristics can enable the design of networks that minimize expected delay.

Expectation

The expected delay until the propagation of a random event is arrested is the sum of the expected delays experienced by each system given the event starts in any of the component systems. Therefore, the expected time that the event "spends" in the network can be defined as:

$$E(F) = \sum_{i} E(F|S_i)P(S_i) \tag{2}$$

where P(S$_i$) is the probability of a random event occurring in system *i* and E(F|S$_i$) is the expected time until the event is arrested given that it started in system S$_i$. This quantity is the sum of all the possible ways the event can reach system *i*. We denote this as E(F$_i$) and define it as:

$$E(F_i) = E(F|S_i) = \sum_{j} E(F_j|T_{ij})P(T_{ij}) \tag{3}$$

where P(T$_{ij}$) is the probability of disruption in system *i* that results in a disruption in system *j*, and E(F$_j$|T$_{ij}$) is the expected time until the event is arrested (expected delay) in system *j* given that it propagated from system *i*. Note that each disruption results in the accumulation of a predetermined amount of delay, D$_{ij}$ and we can write the relationship in (3) as:

$$E(F_i) = \sum_{j} E(F_j + D_{ij})P(T_{ij}) \tag{4}$$

where D_{ij} is the delay in system *i* caused by an event that propagates to system *j*. Because D_{ij} is a constant, we can rewrite this as:

$$E(F_i) = \sum_{j} E(F_j)P(T_{ij}) + D_{ij}P(T_{ij}) \tag{5}$$

This relationship is of the form:

$$\mu = A\mu + b \tag{6}$$

where μ is the vector of the expected time until delay is arrested at each node, A is a matrix of the conditional probabilities P(T$_{ij}$), and *b* is a constant term defined as:

$$b_i = \sum_{j} \delta_{ij} D_{ij} P(T_{ij}) \tag{7}$$

Here, δ_{ij} has a value of 1 when *i* = *j* and 0 otherwise. The matrix A is a transition probability matrix, where each system represents a state (e.g. location of the disrupting event); also it is Markovian, that is, the entries of A are smaller than or equal to one and the rows add up to one. Matrix A is of the form:

$$A = \begin{bmatrix} Q & R \\ 0 & 1 \end{bmatrix} \tag{8}$$

In the context of system development Q represents the probabilities of a disruption propagating to dependent systems while R represents the probabilities of a disruption being arrested (e.g. probability of a disruption going to the final node F and exiting the network). The identity entry represents the final state/node F; the probability of an event being arrested when it is in node F is 1. The matrix Q contains the necessary information to determine the expected time that an event spends in the network given that it is in a given system. The solution to the problem in (6) when solving for μ is:

$$\mu = (I - Q)^{-1}b \tag{9}$$

Because the matrix Q is part of the transition probability matrix A and R is a non-zero vector, in general at least one of the rows of Q sums to a number strictly less than 1 (for the example problem presented here all rows of Q are strictly less than 1). This means that there is one eigenvalue of A that lies on the unit disc and the others are inside the unit disc. This characteristic of A and Q ensure that (I – Q) is always invertible and the eigenvalues of Q are always positive.
The solution to Eq (9) indicates the expected time until a disrupting event is arrested given that it is in any of the systems. Therefore, the total expected delay in (2) is:

$$E(F) - \sum_{i} E(F|S)P(S_i) - \sum_{i} \mu_i P(S_i) \tag{10}$$

For the three-system network in Figure 12**Error! Reference source not found.**, the expected time until a delay is arrested is 3.57 time units, assuming that an event is equally like to hit any of the three systems, the transition probabilities, T^1_{ij}, for the example problem are:

$$T^1_{ij} = A = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 4/5 & 0 & 1/5 \\ 2/4 & 0 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

and the impacts of disruptions, D_{ij} are of one (1) time unit (e.g. one week, one month, etc.).

$$Q = \begin{bmatrix} 1/4 & 1/4 & 1/4 \\ 0 & 4/5 & 0 \\ 2/4 & 0 & 1/4 \end{bmatrix} \quad R = \begin{bmatrix} 1/4 \\ 1/5 \\ 1/4 \end{bmatrix} \tag{12}$$

Additionally, the expected delay if a random event impacts system-1 (μ_1) is 3.43 time units; if it impacts system-2 (μ_2) it is 4.0 time units; and if it impacts system-3 (μ_3) it is 3.29 time units. The expected delay is a metric that describes the entire network and its ability to propagate or arrest delays as well as the criticality of individual systems to propagate delays. In this example, system-2 is the most critical system because a disruption in its development results in the highest expected delay. This reflects the fact that system-2 has the lowest probability of arresting a delay (R(2)=1/5). Conversely, system-3 is less critical than system-1, (a disruption in its development results in smaller expected delay) because, while both systems have the same probability of arresting a delay, there are two systems that depend on system-1 (system-2 and system-3) while there is only one system that depends on system-3 (system-1). Additionally, because disruptions to system-1 can propagate to system-2 (the system with the lowest probability of arresting a disruption) this contributes to disruptions to system-1 resulting in larger expected delays than disruptions to system-3.

Typical measures of network topologies like eigenvector centrality are able to identify this criticality of nodes. Eigenvector centrality is an extension of degree centrality of a node – the number of links connected to a node – that acknowledges that not all links connected to a node are equal (Newman, 2008). In this setting, the importance of links is conveyed by the probability of propagation and the impact of the disruption. While a useful measure, this does not describe the entire network and it cannot be used as a metric to compare different network structures/topologies. The expected delay measure presented here contains information about the network structure (number of nodes and links) as well as the characteristics of the interdependencies. This provides a means to describe the entire network and its performance – in this case, delay propagation.

Network Metric

In order to ensure that the expectation metric contains all the necessary information to compare different networks when their transition probabilities or the impacts of disruptions change, a metric is also needed to serve as a second descriptor of the networks. This must take into consideration the number of nodes and links in a network, the transition probabilities, the impact of disruptions, and the probability of an event occurring in any given system. Here we define the network metric, *M,* as follows:

$$M = \sum_{i}\sum_{j} Q_{ij} b_i P(S_i) \tag{13}$$

Higher interdependency strengths (e.g. larger values for the Q_{ij} and/or b_i entries) mean higher values of *M*. The quantity *M* becomes in this way a means to capture the characteristics of a network and makes possible the comparison of the expected delay when network characteristics change. Higher interdependency strengths (e.g. larger values for the Q_{ij} and/or b_{i} entries) mean higher values of *M*.

Network Comparison

Development of different families of systems can result in the achievement of the same capability-level. Not all families of systems, however, have the same risk characteristics. Each candidate family of systems can be comprised of different systems, different system interdependencies, and different interdependency strengths. An SoS manager may want to decide which solution provides the highest likelihood of success or the smallest expected delays when disruptions impact the development process. We demonstrate the applicability of the metric proposed here to compare networks of systems via an example. Figure 13 presents an alternative family of systems to the systems presented in Figure 12.

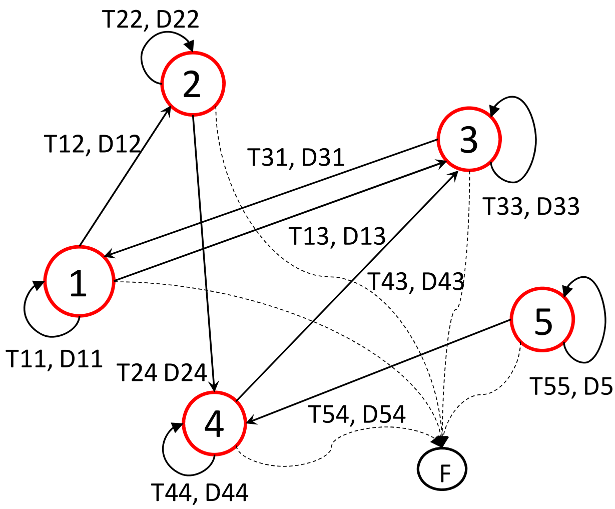


Figure 13. Alternate systems development network

This alternative solution has five systems, which may be different than the three systems of the original alternative, but that provide the same capability. The SoS manager would like to know how the two networks that result from the interdependencies of the different component systems compare when faced with unexpected disruptions.

Assuming that the impacts of disruptions, D_{ij}, are of one (1) time unit and that the probability of a disruption in system *i* propagates to an adjacent system *j*, T^2_{ij}, is:

$$T_{ij}^2 = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 0 & 0 & 1/4 \\ 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 1/3 & 0 & 1/3 & 0 & 0 & 1/3 \\ 0 & 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

the expected total delay is 2.17 time units. Recall that for the three-system network the expected total delay was of 3.57 time units. The five-system network, therefore, is capable of arresting disruptions more effectively than the three-system network, even if it has more systems and interdependencies (seven interdependencies versus four in the three-system network).

Because the impact of a disturbance is of one time unit in both networks, the difference in the expected total delay is due to the number of systems in each network, the number of interdependencies, and their strengths (e.g. the probability of propagation). One way to see this is that the probabilities of an event causing no delay (e.g. probability of going from a system *i* to the sink-node F) are larger for the five-system network (1/4, 1/3, 1/3, 1/3, and 1/3 for system 1, 2, 3, 4, and 5, respectively) than the three-system network (1/5, 1/4, 1/5 for system 1, 2, and 3, respectively). This means that a disturbance has a higher probability of causing no delay, or of being arrested, in the five-system network than the three-system network. While these values are assumptions in this demonstrative example, in actual development networks they can be a result of different system structures, organization, and/or risk profiles.

Impact of disruptions

Here we assume that the transition matrices T^1_{ij} and T^2_{ij} are constant and have the values presented in (11) and (14), respectively, and that the impact of disturbances, D_{ij}, is random (uniformly random between 1 and 10 time units). Differentiating the networks by varying the impact of disruptions helps to demonstrate the ability of the expectation-metric to compare network performance. Figure 14 presents a comparison of the expected total delay in these networks.

Because the impact of disruptions has the same bounds for both networks (uniformly random between 1 and 10 time units), the five-system network always performs better than the three-system network. This points out the importance of interdependencies and their characteristics to the ability of a network to arrest delays.

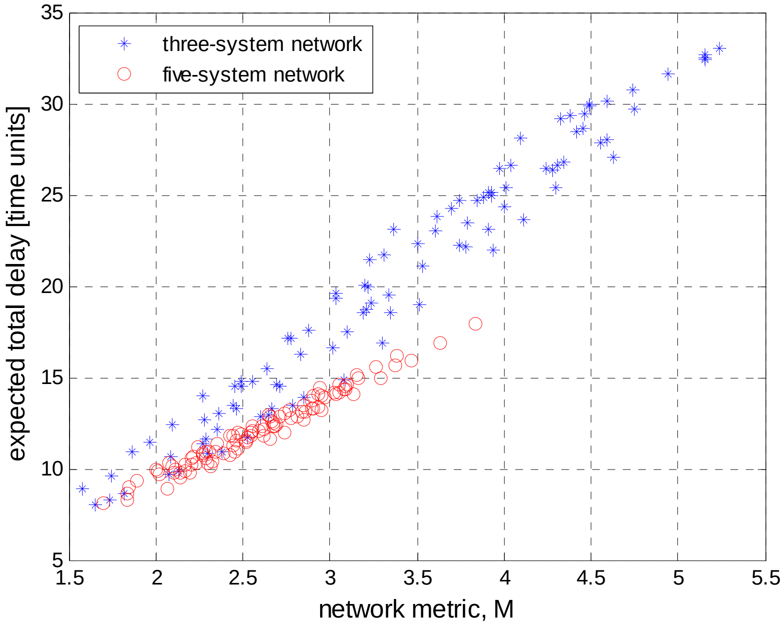


Figure 14. Expected total delay for random impact of disruptions

Impact of network characteristics

To demonstrate the ability of the expectation-metric to capture system, interdependency, and network characteristics we also consider the comparison of networks when the number of nodes, links, and the strength of dependencies (probability of propagation and impact of disruption) varies. Figure 15 presents these trends for random transition probability matrices T_{ij} (ensuring the rows sum to one) as well as random disruption impact matrices D_{ij} (uniformly random values between 1 and 10).

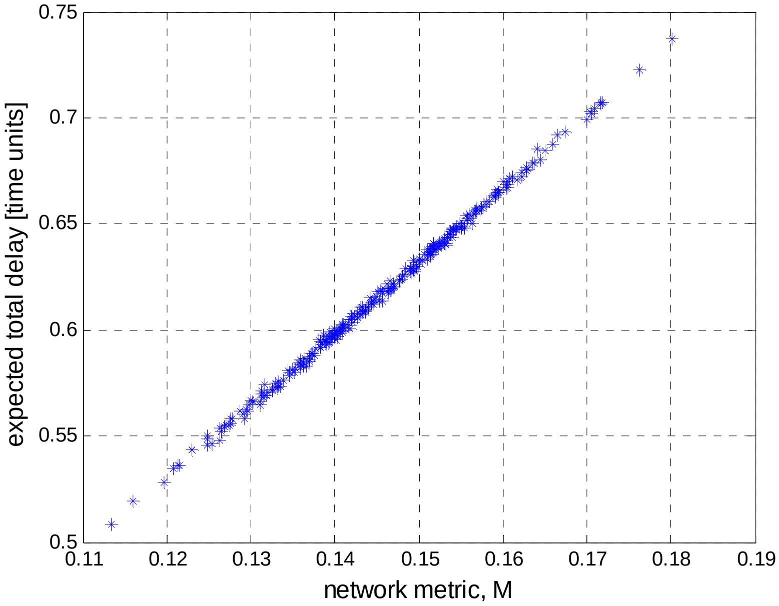


Figure 15. Expected total delay for random transitional probabilities, T_{ij} and random D_{ij}

These trends show that as the network metric *M* increases, the expected delay in the network also increases and the metric *M* is able to capture the pertinent network characteristics. Hence, for a given value of *M* networks can have different number of nodes, links, and different interdependency strengths. Development of interdependent systems has the potential to provide capabilities that go beyond the capabilities of individual systems. The resulting networks, however, introduce new complexities and risk in the development process. Disruptions in the development of one system can propagate and impact the development of other systems. To determine the optimal family of systems that can achieve a desired capability while minimizing the negative impacts of interdependencies requires the quantification of the impact of disruptions and the ability of a network to arrest their propagation within the network.

Overall, the approach can be used to quantify the ability of a network to arrest the propagation of delays that result from such disruptions. Such network-level metric can aid SoS engineers in determining the family of systems that can provide a desired capability while quantifying and, eventually, minimizing the impact of random disruptions throughout the development process. Furthermore, when coupled with simulation-based tools such as the CEM, it can provide a theoretical basis for measuring the performance of the simulation.

Biographies of Investigators

Daniel DeLaurentis is an Associate Professor in the School of Aeronautics and Astronautics Engineering, Purdue University. He received his Ph.D. from Georgia Institute of Technology in Aerospace Engineering in 1998. His current research interests are in Mathematical modeling and object-oriented frameworks for the design of system-of systems, especially those for which air vehicles are a main element; approaches for robust design, including robust control analogies and uncertainty modeling/management in multidisciplinary design.

Muharrem Mane is an Associate Research Scientist in the School of Aeronautics and Astronautics Engineering, Purdue University. He received his Ph.D. from Purdue University in Aerospace Engineering in 2008. His current research interests are in risk analysis and propagation, resource allocation and design under uncertainty, and network modeling and analysis. He currently works in the System of Systems Laboratory led by Dr. DeLaurentis.

References

AhmadBeygi, S., Cohn, A., Guan, Y., Belobaba, P., Sep 2008. “Analysis of the Potential for Delay Propagation in Passenger Airline Networks,” *Journal of Air Transport Management*, Vol. 14, No. 5, pp. 221-236.

Ayyalasomayajula, S., DeLaurentis, D.A., Moore, G.E., Glickman, L.T., Oct. 2008. "A Network Model of H5N1 Avian Influenza Transmission Dynamics in Domestic Cats," *Zoonoses and Public Health*, Vol. 55, Issue 8-10, pp 497-506.

Bar-Yam, Y. 2003. Complexity of Military Conflict: Multiscale Complex Systems Analysis of Littoral Warfare. *New England Complex Systems Institute*. http://www.necsi.edu/projects/yaneer/SSG_NECSI_3_Litt.pdf

Brown, M. and Flowe, R., 2005. “Joint Capabilities and Systems of Systems Solutions: A Multidimensional Approach to Understanding Cost Drivers,” *Defense Acquisition Review Journal*, Vol. 12 No. 2, pp. 138-154.

Butler, J. T., April 2001. “UAVs and ISR Sensor Technology”, Research Report for Air Command and Staff College, Air University, AU/ACSC/033/2001-04, [online source]: https://research.maxwell.af.mil/papers/ay2001/acsc /01-033.pdf [accessed 26 January 2010].

Charles, P. and Turner, P., 2004. “Capabilities Based Acquisition…From Theory to Reality”, CHIPS magazine, [online database]: http://www.chips.navy.mil/archives/04_summer /web_pages/GEMINII.htm [accessed 21 January, 2010].

Committee on Systems Integration for Project Constellation, 2004. “Systems Integration for Project Constellation*”,* The National Academies, [online source] http://nap.edu/html/proj-constella tion/ltr-rep.pdf [accessed 15 January 2009].

Constantine, J. and Solak, S., 2010. “SysML Modeling of Off-the-Shelf-Option Acquisition for Risk Mitigation in Military Programs”, *Systems Engineering*, Vol. 13, No. 1, pp 80–94.

DeLaurentis, D., Han, E-P., Kotegawa, T., 2008. "Network-Theoretic Approach for Analyzing Connectivity in Air Transportation Networks," *AIAA Journal of Aircraft,* Vol.45, No.5, pp. 1669-1679.

Department of Defense Directive 5000.2, 2005. “Chapter 4: Systems Engineering”, Washington DC.

Durkac, L.M., March 2005. “Joint Mission Capability Packages: The Future of Joint Combat”, 10th International Command and Control Research and Technology Symposium, [online database]: http://www.dodccrp.org/events/10th_ICCRTS/CD/papers/063.pdf, [accessed 25 January, 2010].

Fowler, C. A. 1994. The Defense Acquisition System Too Late for the Scalpel; Bring Out the Meataxe! *IEEE Aerospace and Electronic Systems Magazine* 9(8):3-6.

Francis, P., April 2007. “Defense Acquisitions: Missile Defense Agency’s Flexibility Reduces Transparency of Program Cost”, Government Accountability Office’s Testimony before the Subcommittee on Defense, Committee on Appropriations, U.S. House of Representatives, [online source]: www.gao.gov/new.items /d07799t.pdf *[accessed 27 January 2010].*

Gilmore, M.J., April 2006. “The Army’s Future Combat System Program,” CBO Testimony, [online database], http://www.cbo.gov/ftpdocs/71xx/doc7122/04-04-FutureCombatSystems.pdf [retrieved 3 March 2010].

Ghose, S. and D.A. DeLaurentis. 14-15 May 2008. Defense Acquisition Management of Systems-of-systems. *5th Annual Acquisition Research Symposium*, Naval Postgraduate School, Monterey, CA.

Huang, C.D., Behara, R.S., Hu, Q., 2008. “Managing Risk Propagation in Extended Enterprise Networks”, *IT Professional*, Vol. 10, No. 4, pp. 14-19.

Kotegawa, T., DeLaurentis, D.A., Sengstacken, A., Han, En-pei., Sep. 2008. "Utilization of Network Theory for the Enhancement of ATO Air Route Forecast" *8th AIAA Aviation Technology, Integration, and Operations*, Anchorage, Alaska.

Maier, M., 1998. “Architecting Principles for System-of-Systems”, *Systems Engineering* Vol. 1, No. 4, pp. 267-284.

Mane, M. and DeLaurentis,D., May 2009. “Acquisition Management for Systems of Systems: Exploratory Model Development and Experimentation.” *6th Annual Symposium of NPS*, Monterey, CA.

NextGen Integration and Implementation Office. 2009. *NextGen Implementation Plan 2009*. Federal Aviation Administration, [online database] http://www.faa.gov/about/initiatives/nextgen/media/ NGIP_0130.pdf [accessed 25 September 2009].

Newman, M.E.J., 2008. “Mathematics of Networks,” *The New Palgrave Encyclopedia of Economics*, 2nd Edition, L. E. Blume and S. N. Durlauf, Palgrave Macmillan, Basingstok.

Polzer, H., D.A. DeLaurentis, and D.N. Fry. 2007. Multiplicity of Perspectives, Context Scope, and Context Shifting Events. *2007 IEEE International Conference on System of Systems Engineering* :1-6.

Ross, M.S., 2007. “Introduction to Probability Models,” 9th Edition, Academic Press, Burlington, MA.

Rouse, W. 2007. Complex Engineered, Organizational and Natural Systems. *Systems Engineering* 10(3):260-271

Saunders, T.et al., 2005. “Report on System-of-Systems Engineering for Air Force Capability Development”. *USAF Scientific Advisory Board*, SAB-TR-05-04.

Sage, A. and S. Biemer. 2007. Processes for System Family Architecting, Design, and Integration. *IEEE Systems Journal* 1(1):5-16.

Sauser, B., Verna, D., Ramirez-Marquez, J., and Gove, R., 2006. “From TRL to SRL: The concept of Systems Readiness Levels”, Proceedings of the Conference on Systems Engineering Research, CSER, Los Angeles, CA.

Simon, H. 1996. *Sciences of the artificial*. Cambridge, MA: MIT Press.

Spacy, W. L. II, 2004. “Capability-Based Acquisition in the Defense Agency and Implications for Department of Defense Acquisition”, *Journal of Contract Management*, pp. 10-19.

Spring, B. 2005. Congressional Restraint Is Key to Successful Defense Acquisition Reform. *The Heritage Foundation*. http://www.heritage.org/Research/NationalSecurity/bg1885.cfm (March 26, 2008).

U.S. Army. 2008. Future Combat Systems. https://www.fcs.army.mil/

U.S. Department of Defense, 2008a. Defense Acquisition Guidebook. [online database]: https://akss.dau.mil/ dag/ [accessed 4 April 2008].

U.S. Department of Defense, 2008b. “Systems Engineering Guide for System-of-systems”, [online database]: http://www.acq.osd.mil/sse/ssa/initiat_sos-se.html\ [accessed 2 March 2008].

U.S. Government Accounting Office. *Issues facing the Army's Future Combat Systems.* ONLINE GPO Access. 13 August 2003. Available: http://www.gao.gov/new.items/d031010r.pdf

U.S. Government Accountability Office. *Coast Guard: Observations on Agency Performance, Operations and Future Challenges*. June 2006. Available: http://www.gao.gov/new.items/d06448t.pdf

U.S. Government Accounting Office. *Future Combat System Risks Underscore the Importance of Oversight*. ONLINE. GPO Access. 27 March 2007. Available: http://www.gao.gov/new.items/d07672t.pdf

U.S. Government Accounting Office. *Role of Lead Systems Integrator on Future Combat Systems Program Poses Oversight Challenge.* ONLINE. GPO Access. June 2007. Available: http://www.gao.gov/new.items/d07380.pdf

Xu, N., Donohue, G., Laskey, K.B., Chen, C.H., June 2005. "Esti- mation of Delay Propagation in the National Aviation System Using Bayesian Networks," 6th USA/Europe Air Traffic Management Research and Development Seminar.

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org