SYM-AM-16-053

PROCEEDINGS
OF THE
THIRTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

THURSDAY SESSIONS
VOLUME II

Modeling Uncertainty and Its Implications in Complex Interdependent Networks

Anita Raja, Professor, The Cooper Union
Mohammad Rashedul Hasan, Assistant Professor of Practice, UNL
Robert Flowe, Office of Acquisition Resources & Analysis, OUSD (AT&L)
Brendan Fernes, Student, The Cooper Union

Published April 30, 2016

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

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Panel 13. Setting Requirements and Managing Risk in Complex, Networked Projects

Thursday, May 5, 2016	
9:30 a.m. – 11:00 a.m.	**Joseph Yakovac, Lieutenant General, USA (Ret.)**– Former Principal Military Deputy, Assistant Secretary of the Army for Acquisition, Logistics and Technology ***Acquisition in a World of Joint Capabilities: Methods for Understanding Cross-Organizational Network Performance*** Mary Brown, Professor, UNCC Zachary Mohr, Assistant Professor, UNCC ***Modeling Uncertainty and Its Implications in Complex Interdependent Networks*** Anita Raja, Professor, The Cooper Union Mohammad Rashedul Hasan, Assistant Professor of Practice, UNL Robert Flowe, Office of Acquisition Resources & Analysis, OUSD (AT&L) Brendan Fernes, Student, The Cooper Union ***An Optimization-Based Approach to Determine System Requirements Under Multiple Domain-Specific Uncertainties*** Parithi Govindaraju, Graduate Research Assistant, Purdue University Navindran Davendralingam, Research Scientist, Purdue University William Crossley, Professor, Purdue University

Modeling Uncertainty and Its Implications in Complex Interdependent Networks

Anita Raja—is the Associate Dean of Research and Graduate Programs and Professor of Computer Science in the Albert Nerken School of Engineering at The Cooper Union. She directs the Distributed Intelligent Agents Lab. She received her PhD in computer science from the University of Massachusetts at Amherst in 1998 and 2003, respectively. Raja's research focus is in the field of artificial intelligence, specifically as it relates to the study of decentralized control and reasoning in software agent systems operating in the context of uncertainty and limited computational resources. [araja@cooper.edu]

Mohammad Rashedul Hasan—is an Assistant Professor of Practice in the Department of Computer Science and Engineering at the University of Nebraska–Lincoln. He received his PhD in Computing and Informatics from the University of North Carolina at Charlotte in 2014. His PhD adviser was Anita Raja. He obtained a Master of Applied Information Technology degree from Monash University (Australia) under the Australian Government Development Scholarship. He was the entrepreneur of a non-profit IT start-up company named Global Village.com that provided telecommunication services to the rural people in Bangladesh. Hasan's research interest revolves around artificial intelligence, network science and complex systems. [hasan@unl.edu]

Robert Flowe—is a Senior Program Analyst supporting the Deputy Director Office of the Secretary of Defense Studies (OSD) and Federally Funded Research and Development Center (FFRDC) Management, conducting policy development and oversight of DoD FFRDCs and University Affiliated Research Centers (UARCs). He facilitates the proper utilization and management of DoD FFRDCs and UARCS in accordance with relevant laws, regulations, and policies, and enhances the effectiveness of these strategic resources in support of DoD goals. [robert.m.flowe.civ@mail.mil]

Brendan Fernes—is an undergraduate student in engineering at The Cooper Union. He is in his sophomore year. [fernes@cooper.edu]

Abstract

The overall goal of this paper is to continue our efforts to forge new ground in identifying the effects of interdependencies in large complex networked applications and, if needed, uncovering early indicators of interdependency risk so that appropriate risk mitigation actions may be taken. Specifically, we seek to study and quantify the impact of network characteristics on cascading risk. Cascading risk is defined as the propagation of programmatic issues across networked programs due to the interdependency of one program upon the other. Harnessing the extensive data that has been collected over the years in the form of Defense Acquisition Execution Summary (DAES) and Selected Acquisition Reports (SARs) documents for Major Defense Acquisition Programs (MDAPS), we will present our intermediate results in our ongoing efforts on leveraging network structure and sequential data to study cascading risks. We will also identify the challenges to data acquisition.

Introduction

Our work is motivated by the need for "what-if" analysis in large complex interdependent and networked applications such as the critical infrastructure network (electric, water, gas grids). The research goal is to develop methodologies and algorithms to proactively model and reason about non-linear cascading risks to facilitate this analysis. Networked applications often operate under uncertainty in environmental response and the temporal state and action choices of the nodes are captured in the form of structured and unstructured text data as well as image data.

We build on our previous work (Raja, Hasan, & Brown, 2012; Raja et al., 2013, 2014), where we used state-of-the-art extraction technologies including Latent Dirichlet

Allocation (LDA) topic modeling algorithms to develop automated text and image extraction techniques to extract features from various types of structured and unstructured text and image data. In addition to this automated data extraction module, we developed two executable modules: one to identify the relationships in a network (Network Identifier module) and the other to compute the weight of the links among the neighboring nodes (Interdependency Index Determiner). We tested and evaluated these algorithms on a small network and showed that the performance of the automatic extraction algorithms was comparable to the performance of manual extraction.

We use the MDAP network as a case study to study cascading risks and develop methodologies and algorithms that can be generalizable to similar networks. Individual MDAP performance across months and years has been captured by a combination of structured and unstructured temporal data, including Selected Acquisition Reports (SARs), Defense Acquisition Execution Summary (DAES) reports, and milestone reviews are evaluated from an individual program point of view without emphasizing the dynamics of joint space. The question of modeling cascading risk across programs with funding or data relationships is important since we conjecture that poor performance of the MDAPs (various breach conditions) can be attributed to local (individual MDAP) as well as non-local (related MDAPs) sources that result due to interdependencies among the MDAPs.

In this paper, we present a network-centric approach that has the dual goal of contributing to advances in reasoning about uncertainty, large-scale text and image data analysis, as well as understanding of complex networks. This project breaks ground in the areas of (a) defining a metric to quantify the influence of network characteristics on performance and (b) identifying the type of data required to formulate appropriate mathematical models for understanding the dynamics of complex networks.

Network Performance Study From an Interdependent Hierarchical Network Perspective

The joint space of major defense acquisition programs (MDAPs) creates interdependencies among MDAPs. These interdependencies contain the characteristics of a complex network (Brown, 2014). Programs in the MDAP network share diverse relationships. Mainly, there are two types of ties that exist among the MDAPs: (1) programmatic ties (also called programmatic interdependencies) are defined by the program managers in terms of inbound and outbound connections to support hardware/software requirement of the programs, and (2) funding ties that identify the programs as funding neighbors if they draw funding support from the same “program element” (PE) account. These two types of ties result in two types of network relationships among the MDAPs, namely, programmatic network and funding network.

A systemic understanding of the performance of the MDAPs requires the understanding of these two types of networks. Therefore, the system of MDAPs can be considered as a multiplex network that is a superposition of both programmatic and funding networks defined on the same set of programs (Szell, Lambiotte, & Thurner, 2010).

Interdependencies among the program influence the performance of the MDAPs (Brown, 2014; Raja et al., 2012). However, the multiplex nature of the MDAP network has not been considered to examine the performance of the programs. Moreover, the effect of interdependency on the programs was not quantified previously. Our goal is to investigate the joint space of the MDAP multiplex network as it influences program performance and to define a metric (the risk parameter) that quantifies this influence. The values of this risk parameter for each program in the multiplex network would be useful to forecast a potential cascading effect. Moreover, the program managers would be able to identify critical programs using this parameter and take necessary measures to improve programs' performance. The risk parameter is formally defined in the following section based on the Probabilistic Risk Analysis (PRA) methodology for networked systems.

Probabilistic Risk Analysis (PRA)

Probabilistic risk analysis (PRA) is a methodology (Lewis, 2009) to evaluate risks associated with a complex engineering entity. It systematically looks at how the pieces of a system work together to ensure safety. PRA allows analysts to quantify risk and identify what could have the most impact on safety (Lewis, 2009). Therefore, we use the risk parameter from PRA methodology to quantify the influence of interdependency in a complex network, specifically the MDAP network.

The PRA equations for risk in a system use the notion of vulnerability and consequence. Although the concept of vulnerability, risk, and consequence in non-network systems share standard definitions in financial and engineering communities, these terms are not well understood for networked systems. This is because network science is a new field, and it is not very clear how to understand the failure of the assets in networks.

According to the standard definitions (Lewis, 2009) in non-networked systems, vulnerability V is the probability that a component or asset will be compromised after successful attacks. Risk R measures the expected loss due to the failure of an asset. Threat T is the probability that an attack will be attempted. Consequence C is the outcome of a successful attack. Therefore, standard risk is defined as the product *R=TVC.*

These definitions, however, do not provide an appropriate measure for risk in networked systems. In a network, system failure is a function of the interdependence of the nodes. These definitions do not incorporate the interdependency of the various components of a system. Therefore, it is important to consider the connectivity among the nodes in a network for computing risk.

Lewis (2009) extended these standard definitions to networks containing many components or assets (nodes and links). Threat (t), vulnerability (v), consequence (c), and risk (R) in a networked system are an aggregation of individual component or asset threat, vulnerability, and consequences. Network risk is defined in the following PRA equation as an expected value by taking the sum over all nodes (n) and links (m) of the individual components: $\boldsymbol{R} = \sum_{i=1}^{n+m} \boldsymbol{t_i} \boldsymbol{v_i} \boldsymbol{c_i} = \sum_{i=1}^{n+m} \boldsymbol{v_i} \boldsymbol{c_i},$ assuming $t_i = 1$. Here, threat and vulnerability are a priori estimates of the probability of failure. Consequence is typically measured in dollars or lives. This PRA equation for risk is applicable for any system where a priori approximations of the probability of failure can be reasonably estimated. For reducing risk in a networked system, Lewis (2009) argues that it is important to identify the critical nodes that have higher risk values.

Earlier works by Albert, Jeong, and Barabasi (2000) and others explored why highly-connected nodes were more critical nodes than others. However, these studies were done in the context of a single-plex network. Al-Mannai and Lewis (2007) proposed a static technique for critical node analysis in a **multiplex network** where criticality of a node not only depends on the number of connections, but also on other measures. They use a degree-weighted model of network risk to identify the most critical nodes in a network. Intuitively, critical nodes either have many connections or have larger target values. Based on this observation, Al-Mannai and Lewis (2007) extended the simple PRA definition of risk to define the target value of a node as g_iC_i , where g_i is the degree of the node and C_i is the consequence associated with the node's intrinsic value. Therefore, according to their model, extended risk r for an n-node network is related to network topology as follows:

$$r_{ext} = \sum_{i=1}^{n} g_i V_i C_i \quad (1)$$

where g is the degree of node i, while V and C are its vulnerability and consequence, respectively.

Example: PRA for a Small Synthetic Network

As an illustration of the above-mentioned extended PRA technique, let's consider the following network (Figure 1) of four nodes (A, B, C, and D). Connectivity among the nodes is shown for three years. We will use fictitious values for vulnerability and consequence of the nodes in this network in order to understand how the above-mentioned model helps to identify nodes that are most critical for the operation of the network. Also, it will facilitate in understanding the various factors that contribute towards the criticality measure.

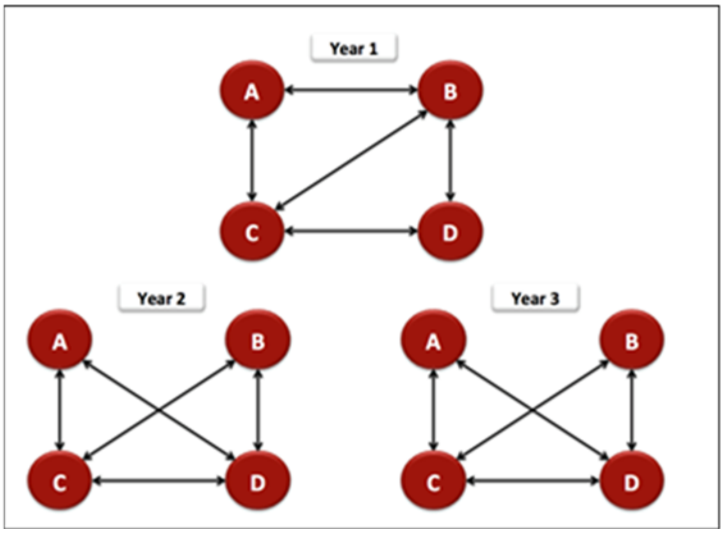


Figure 1. Critical Node Analysis for a Synthetic Network

Table 1 shows the results for extended PRA. In Year 1, we notice that node C is the most critical. Although it has the smallest consequence value in the network, its high connectivity and largest value for the vulnerability are responsible for its critical condition.

In Year 2, however, node C is not the most critical node anymore. This is due to the reduction in its consequence measure. Node D appears to be the most critical because of the increase in its degree. Its vulnerability and consequence values did not increase from the previous year.

In Year 3, Node A becomes the most critical node because of the increase in its vulnerability and consequence values. However, its degree did not increase.

Table 1. PRA of the Synthetic Network for Three Years

Year 1

	g	V	C	r = gVC
A	2	0.01	15	0.3
B	3	0.02	20	1.2
C	*3*	*0.6*	*10*	***18***
D	2	0.3	15	9

Year 2

	g	V	C	r = gVC
A	1	0.07	10	0.7
B	2	0.01	30	0.6
C	3	0.5	2	3
D	3	0.3	15	***13.5***

Year 3

	g	V	C	r = gVC
A	1	0.6	20	***12***
B	2	0.01	30	0.6
C	3	0.5	7	10.5
D	3	0.1	10	3

Figure 2 shows the change in extended risk values for the nodes that indicate node criticality during the three-year time-span. This simple illustration helps us to understand the significance of incorporating a node's degree (g) for the computation of its risk along with its vulnerability and consequence (Al-Mannai & Lewis, 2007).

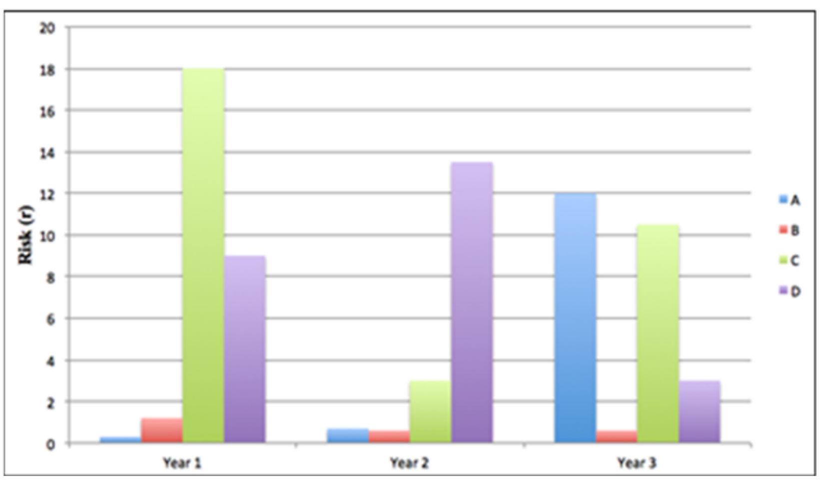


Figure 2. Critical Node Analysis for the Synthetic Network

Critical Node Analysis for a Small MDAP Network

Implementing the above-mentioned technique of extended PRA is a non-trivial task for MDAP networks. PRA requires a reasonable estimation of a priori approximations of vulnerability and consequence of the network assets. However, there is no guideline to do such estimation for MDAPs. Moreover, data on the MDAPs are complex artifacts and often times are either incomplete or fuzzy. Therefore, defining vulnerability and consequence parameters for MDAPs is a challenging task that we address below.

MDAPs operate on a multiplex network. At one hand, MDAPs share funding with other MDAPs (as a result, they form a shared-funding network); on the other hand, MDAPs share hardware/software components with other MDAPs (as a result, they also belong to a programmatic network). Therefore, performance of MDAPs can be examined in light of the performance of the individual program (program-centric) as well as its resulting performance in two different networks (network-centric): (1) a **programmatic network** and (2) a **funding network.** In our analysis, we consider both the program-centric and network-centric contributions.

Below, we first discuss how to discover diverse (programmatic and funding) network relationships among the MDAPs and form a multiplex network. Then we define the various parameters for the extended PRA model. Finally, to validate the approach for extended PRA of the MDAP network, we present a case study of critical node analysis for an MDAP enterprise.

Multiplex Network Formation

The interdependency of the MDAPs that influence their performance can be best understood via the programmatic network (Brown, 2014). In a programmatic network, individual MDAPs support other MDAPs by providing software or hardware components. Therefore, our network of interest is based on the programmatic relationships that exist among the MDAPs. We have gathered data on programmatic interdependencies from the DAES reports for the respective MDAPs. Typically, the last page of the DAES report records the inbound and outbound connections.

Apart from their programmatic dependency, the MDAPs are also related via common PE accounts. In our network model, we capture this funding network relationship as well.

Both the programmatic and funding relationships on the same set of MDAPs are superimposed to define a multiplex MDAP network. For example, Figure 3 shows both the funding and programmatic interdependencies among the MDAPs in an MDAP multiplex network in 2009.

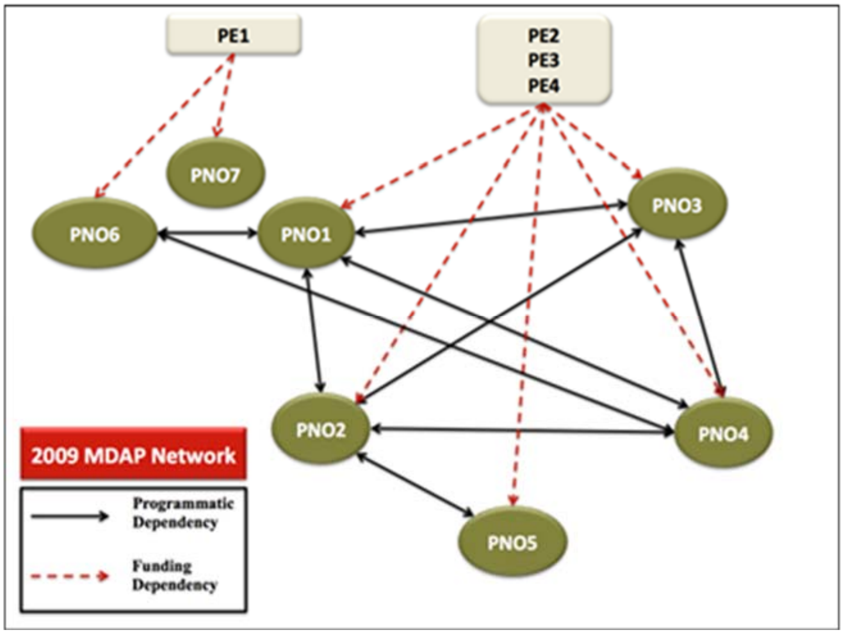


Figure 3. An MDAP Multiplex Network Model

Parameters of Extended PRA Model: Degree (g), Vulnerability (V), and Consequence (C)

We define the extended PRA model parameters as follows:

- **Degree (g):** It is defined by the number of outgoing edges from a node in the programmatic network. Therefore, degree measures the extent of influence of one program (node) on other programs. In an m-node network,

$$g = \sum_{i=1}^{m} n_i \tag{2}$$

- **Vulnerability (V):** It is a measure of weakness of a node in a network. It is defined as the probability of failure of a node if a successful attack is launched on it.

In the MDAP network, we notice that a program may become prone to failure; we call such a program a critical program. Our hypothesis is that breach incidences and other factors mentioned below are indicators of criticality of a program. Program failure is characterized by increased APB breaches and PAUC increase. Moreover, we hypothesize a program's criticality could potentially influence its neighbor's performance (increased breached condition and PAUC increase).

- First, from a program-centric point of view, a program may fail due to its intrinsic poor performance. For example, a weapons procurement cut could lead to intrinsic poor performance (Raja et al., 2012). This program-centric view is captured in the "Program Status" page of the DAES report of the programs.
- Second, APB beaches and the percentage of increase in PAUC is also a measure of a program's performance. These values are recorded in SAR files.

- Third, from a program-centric perspective, the number of its funding and programmatic neighbors influences the performance of a MDAP. For instance, having a large number of funding neighbors (per PE account) makes a program susceptible to potential reduction in promised funding as funds could be siphoned to its neighbors.
- Fourth, having a large number of upstream programmatic neighbors (from which the edges fall on the program) increases its dependency of software/hardware components for successful completion of its tasks.
- Fifth, funding lag also affects the performance of a program and may make it prone to failure.

We propose that the above-mentioned five parameters provide a reasonable estimation for the probability of failure of a program and use these to define vulnerability (Lewis, 2009). Therefore, vulnerability should be considered as the cumulative effect of these parameters. We define the normalized vulnerability based on these parameters using a simple linear function and study its effectiveness:

$$V=\frac{p+b+fNbor+pNbor+diffF}{1+1+1+1+1} \tag{3}$$

Each parameter in the numerator has a maximum value of 1. In the following, the individual parameters are formally defined.

p: It refers to a program's intrinsic performance (captured in DAES reports) and is a linear combination of the factors contributing to Program Status. We use the December DAES report for the last reported month of a year for this computation. The last reported month's data is used as it provides that year's intrinsic performance level of the program. We use the data provided in the "Program Status" page of the DAES reports to compute this metric as described in Table 2.

b: It refers to the number of breaches that occurred in the current year (retrieved from SAR files).

fNbor: It is the normalized number of funding neighbors (retrieved from R docs).

pNbor: It is the normalized number of upstream programmatic neighbors (retrieved from DAES reports).

diffF: It is the normalized differential between received and promised funding amounts (retrieved from SAR and R docs).

Table 2 reports the formulas that we defined to compute these five parameters.

Table 2. **Formulas for the Five Parameters Used in the Computation of Vulnerability**

Parameters	Formula
p	$$\frac{Cost + Schedule + Performance + Funding + Life\ Cycle\ Sustainment}{10+10+10+10+10}$$ For the five "Program Status" variables, Cost, Schedule, Performance, Funding and Life Cycle Sustainment, we map the following quantitative values for the colored bubbles: Green: 0; Yellow: 5; Red 10 The value of p is normalized by the maximum numeric values (i.e., 10) for each status variable.
b	$$\frac{APB\ SchedBreach + APB\ PerfBreach + APB\ Cost\ (RDT\&E) Breach + PAUC}{1+1+1+25}$$ where APBSchedBreach = 1 if APB Schedule Breach occurred, 0 otherwise; APB Cost (RDT&E) Breach = 1 if APB Cost (RDT& E) Breach occurred, 0 otherwise; APB PerfBreach if APB Performance Breach Breach occurred, 0 otherwise **PAUC**: It is captured from the "Unit Cost" section of the SAR. We use the Current year value. Since a "critical" Nunn-McCurdy breach occurs when the program acquisition or the procurement unit cost increases 25% or more over the current baseline estimate, we use 25 as the maximum value for PAUC.
fNbor	$$\frac{\sum_{i=1}^{n} fNbor_i}{fNbor\_Max}$$ Here, the subscript i refers to each PNO account, and fNbor_Max is a predefined large value that is used for normalizing fNbor. We define fNbor_Max as follows, fNbor_Max = Total PE accounts in the network * Total number of MDAPs
pNbor	$$\frac{Number\ of\ upstream\ programmatic\ neighbors}{pNbor\_Max}$$ pNbor_Max is a predefined large value that is used for normalizing pNbor. We define pNbor_Max as follows: pNbor_Max = Total number of MDAPs in the network
diffF	$$\frac{Promised\ Funding - Received\ Funding}{Promised\ Funding}$$

- **C (Consequence):** Consequence measures the damage or loss (in dollars) of an asset when failure occurs. Therefore, it should be proportional to the RDT&E funding (from R Docs) and is determined by the breach condition. For example, if a program experiences 100% breach, then its Consequence would be tantamount to its entire RDT&E funding. We define it as follows:

$$C = b^{*} \text{ Funding (RDT\&E)} \tag{4}$$

The breach parameter b from the vulnerability computation is used to compute Consequence.

Case Study: An MDAP Network

We use the extended PRA to identify the most critical nodes for an MDAP enterprise that consists of six MDAPs: PNO1, PNO2, PNO3, PNO4, PNO5, and PNO6. These six MDAPS are funded by four program elements (funding sources): PE1, PE2, PE3, and PE4, as shown in Figures 4–6.

Data for the years 2009 to 2011 are used for this case study. Figures 4–6 show the MDAP enterprise multiplex network for these three years.

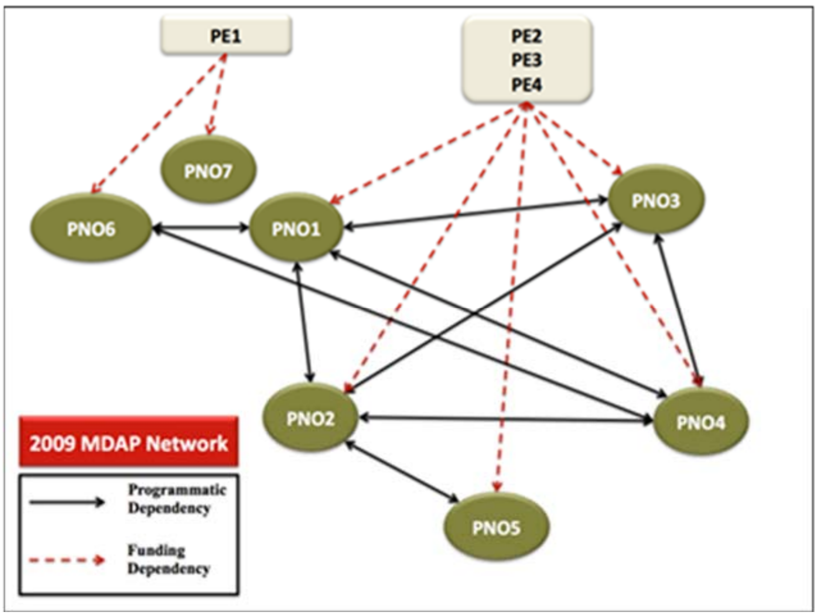


Figure 4. The MDAP Enterprise Multiplex Network, 2009

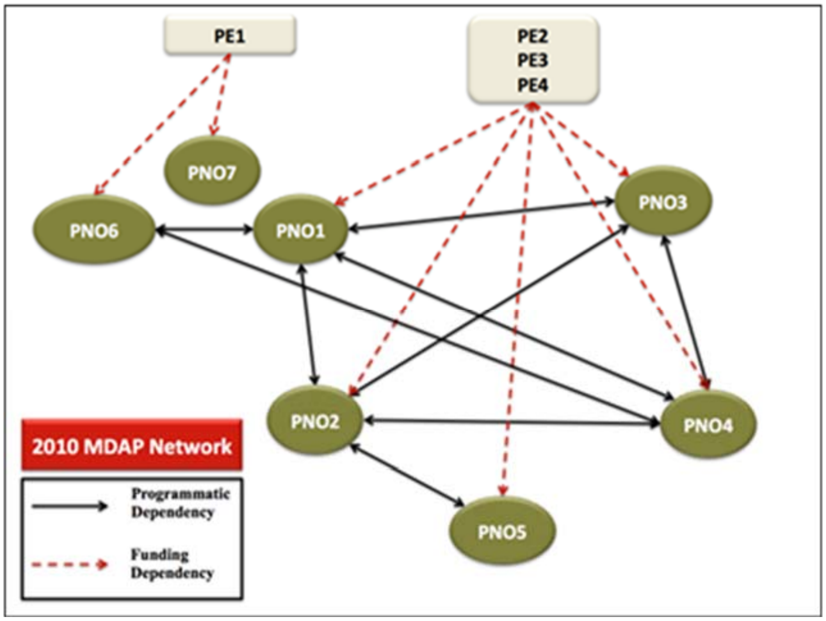


Figure 5. The MDAP Enterprise Multiplex Network, 2010

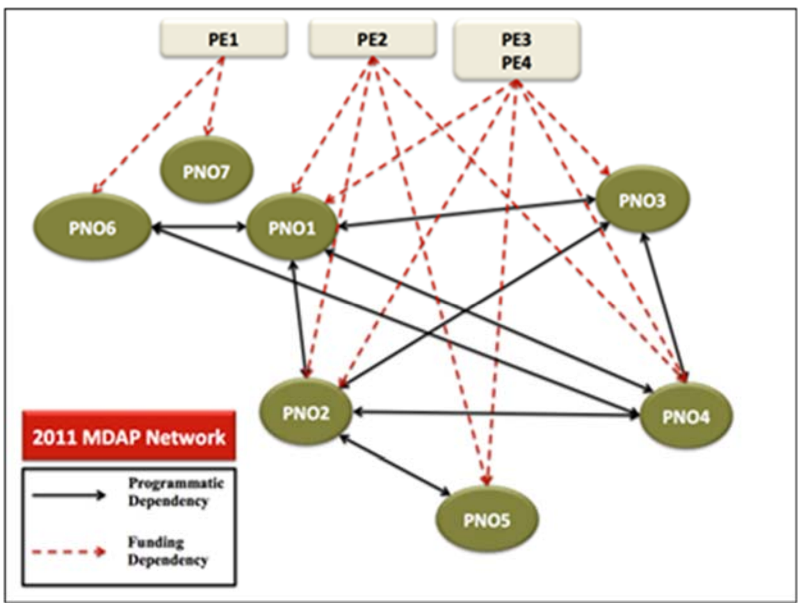


Figure 6. The MDAP Enterprise Multiplex Network, 2011

Detailed calculation of the risk values for each MDAP for three years was performed. Table 3 shows the summary of the results.

Table 3. Critical Node Analysis for MDAP Enterprise Network

	Risk (r)		
	2009	**2010**	**2011**
PNO1	18.11	21.02	97.05
PNO2	0	6.97	3.75
PNO3	6.16	18.89	100.52
PNO4	4.88	11.39	0
PNO5	0	0.17	0.48
PNO6	24.22	24.9	19.7

As an illustration of the calculations in Table 3, we show the detailed calculation of the risk value for PNO1 in 2009 in Table 4.

Table 4. Detailed Calculation of the Risk Value for PNO1 in 2009

Parameters	Calculation
p	$\frac{Cost + Schedule + Performance + Funding + Life\ Cycle\ Sustainment}{10+10+10+10+10}$ $= \frac{0+0+0+0+0}{10+10+10+10+10} = 0$
b	$\frac{APB\ SchedBreach + APB\ PerfBreach + APB\ Cost\ (RDT\&E) Breach + PAUC}{1+1+1+25}$ $= \frac{0+0+0+2.45}{1+1+1+25}$ $= 0.0875$
fNbor	$\frac{\sum_{i=1}^{n} fNbor_i}{fNbor\_Max}$ $= 12/28 = 0.429$
pNbor	$\frac{Number\ of\ upstream\ programmatic\ neighbors}{pNbor\_Max}$ $= 4/6 = 0.6667$
diffF	$\frac{Promised\ Funding - Received\ Funding}{Promised\ Funding}$ $= \frac{215.934(0604280N) - 212.6(SAR)}{215.934(0604280N)}$ $= 0.015439903$
Vulnerability (V)	$V = \frac{p+b+fNbor+pNbor+diffF}{1+1+1+1+1} = 0.2396356$
Consequence(C)	$C = 18.894225$
Risk(R)	$gVC = 18.11091575$

From Figure 7, we observe that over the years, PNO1 and PNO3 became the most critical programs in the network. PE6 retained its criticality level, and we do not see significant improvement. A careful analysis of the data for PNO1 and PNO3 in year 2011 reveals that both programs have high breach incidence (that includes increased PAUC). As a result, their consequence values increased as well. Also, these two programs were characterized by higher degrees. All these factors contributed to their high level of criticality. For PNO6, although its degree is relatively small, it has been experiencing schedule and cost breaches as well as increases in PAUC for three consecutive years. The funding budget for PNO1 and PNO6 (over $300 million) is also a contributing factor.

According to 2011 SAR files, PNO1, PNO3, and PNO6 experienced significant PAUC increase and APB breaches, indicating their poor performance level. This observation confirms that our risk computation measure is a step in the right direction.

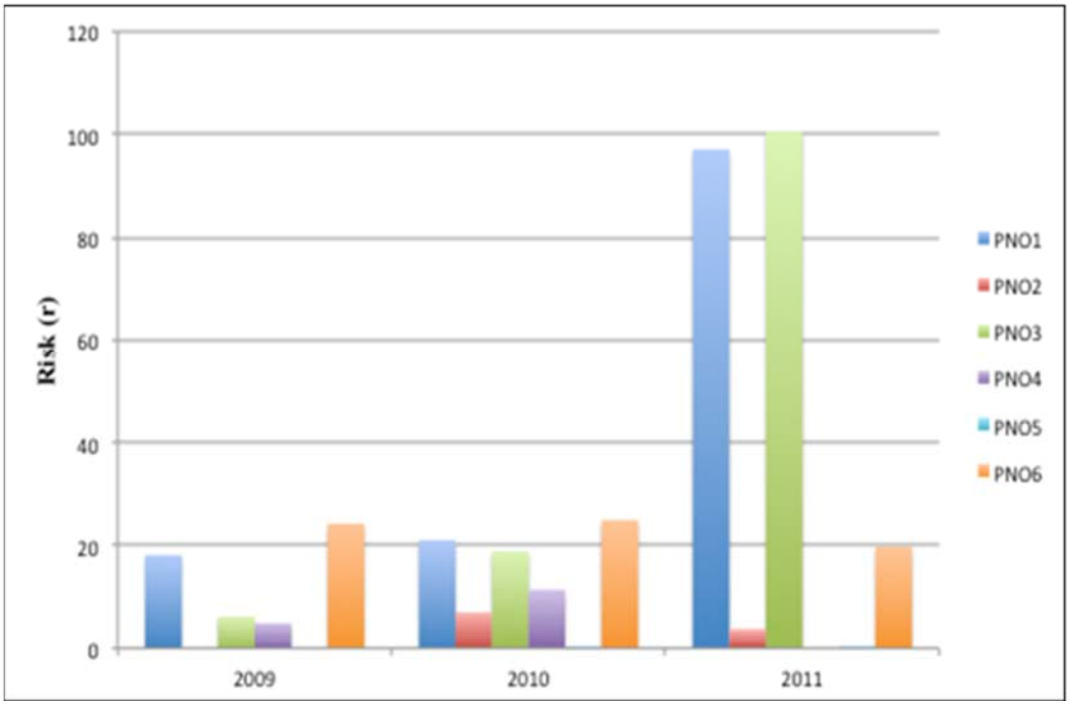


Figure 7. Critical Node Analysis for the MDAP Enterprise Network

Discussion of Extended PRA Model for Risk Computation

The objective for defining the risk parameter (R) in this paper is to capture the effect of multiplex network relations on an MDAP program's performance. As mentioned earlier, breach conditions (from DAES and SAR) are indicators of a program's intrinsic performance, but do not account for the exogenous effects on a program. We have developed the PRA risk model with the potential to capture the network effect on a program's performance. In this model, the intrinsic parameters (p and b) tell us whether a program is "Vulnerable," while the MDAP program's network status accounts for "Criticality." The premise shown by the case study is that this criticality measure helps identify programs that are susceptible to future breaches more effectively than by simply using their intrinsic performance parameters (p or b values).

Manual analysis of MDAP data (done in previous phases of the project) facilitated the process of modeling a small MDAP network for extended PRA analysis. For this modeling, we considered the multiplex nature of the MDAP network and used various performance reports. The results indicate that the extended PRA technique has the potential to successfully identify risky programs and infer the performance of programs.

Also, the PRA analysis uses a network-based composite metric, instead of just individual program PAUC increase and APB breaches, to compute the risk level of a program. For example, both PNO1 and PNO3 have relatively high degree and share the same funding accounts, which makes them susceptible to poor performance. By looking at their increasing PRA risk values in 2009 and 2010, it can be inferred that these two programs are in critical condition. Also by looking at the nearly stable high-risk values of PNO6 in 2009 and 2010, this program should be considered critical as well.

Hence, with the aid of an automated information retrieval mechanism from the performance reports, it is possible to develop an algorithmic tool to identify risky programs. Recognizing the potential of these risky/critical programs to affect the performance of their neighbors could contribute towards predicting cascading effects. As future work, we plan to verify this empirically.

Also, we plan to use this model on another MDAP network to determine if it is able to identify the critical programs; we will modify the parameters of our PRA based model (if necessary) and use this knowledge to define a general model for the entire MDAP network as whole or more realistically, specialized PRA models for classes of similar MDAPs.

Studying the Feasibility of Mathematically Modeling the Phenomenology of MDAP Networks

We also conducted a feasibility study for modeling interdependent networks as a coupled dynamical system and potentially adapting the algorithms for feed-forward networks (Mintchev & Young, 2009; Lanford & Mintchev, 2015) to risk propagation interdependent networks like the MDAP network.

To do this, we would have to determine the network model which includes determining network architecture properties, including various centrality measures, strength of network connections including a precise form of the coupling formalism (strength can be seen as a precise rule that determines dynamical evolution), state features and action options which were already determined in Raja et al. (2012), and a reward optimization model that provides some dynamics to this network. The model would allow us to investigate whether the system has any attractive equilibria, as well as determining the strengths and weaknesses of the basins of attraction. For example, if the steady state of the MDAP network is characterized by only one funded program, with all others having discontinued funding, this is probably not good. We hypothesize that if good equilibria were discovered, an outcome of this analysis could be to recommend a funding strategy that maintains equilibrium or guarantees a rapid convergence toward it.

The specific working hypothesis in the context of the MDAP network is as follows: The programmatic interdependencies between MDAPs have a profound influence on large-scale network performance over an extended period of time.

To determine a network model that is descriptive, predictive, and mathematically sound, we would need a collection of numerical quantities either measured or somehow computed from other measurements recorded in a time series over a sufficiently long period of time. This would involve

- (R1) determination of observable quantities measured numerically (i.e., real numbers on a well-defined scale). The KEY characteristic is to have some a priori evidence that the observables chosen evolve dynamically (i.e., change over time); also, it is absolutely NECESSARY for these to be numerical, or to correspond to some sort of real number scale.
- (R2) finding time series of data on the observables chosen in (R1). Usually a lot of data over a sufficiently long time scale is required to build this historical account of how the observables have changed over time. If the model is to be predictive in the short term, then the variables/observables must have been sampled at a sufficiently high rate.

Evaluating DAES Data

We began by studying the DAES data of several MDAPs collected over a decade with the hope that the sequential monthly data would provide indicators of performance degradation. We have extensive experience with DAES data from our previous work, where we used DAES data to study local and non-local issues that affect the performance of the

MDAP (Raja et al., 2012) and also developed sophisticated text and image extraction tools (Raja et al., 2013, 2014) to automatically extract the DAES data en masse.

Since changes in total cost could be considered as a useful observable, we constructed a few test time series based on the information captured on Top Cost Drivers in the DAES report. Figure 8 captures one such example. It became clear the cost driver time series was not sufficiently volatile enough to facilitate predictability.

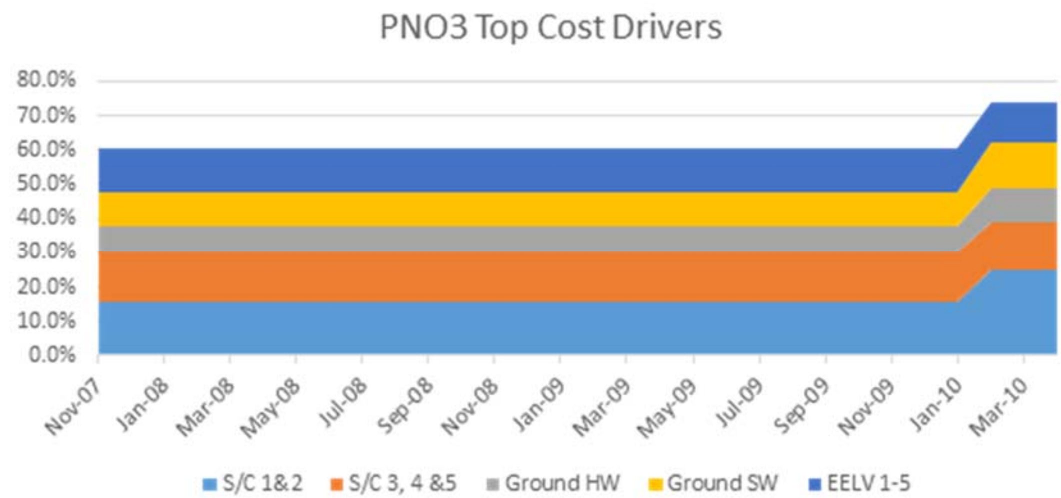


Figure 8. Stacked Area Time Series Data of Five Top Cost Drivers of PNO3

While there is some volatility in January 2010, the volatility is not frequent enough to capture the change in performance risk of the MDAP program over time.

Moreover, we ran into several challenges with the preciseness of the data as far as our goal of building a mathematical model is concerned. Some of our observations are captured below:

In the DAES Program Status page,

- We could not ascertain the quantitative mechanism for color transitions of the red, yellow, or green bubbles that capture the changes in value of Cost, Schedule, Performance, etc. in going from one month to the next.
- The risk in the Risk Summary page describes the risk computation in somewhat of a quantitative way. However, it was still unclear how the risk quantity evaluated; it seems to be coded by a 2-dimensional vector, a (consequence, likelihood) pair; how (if at all) is each of those coordinates computed?

In the DAES page with Top Cost Drivers, Technology Readiness Assessment, Performance (kpps), and Acquisition Program Baseline (APB),

- All of the KPP diagrams seem to be set at T (or threshold); it was not possible to ascertain how these quantities were computed and whether they change over time.
- While the technology readiness assessment box is another potentially interesting measure with regards to building a state space model, the values

did not change for long periods of time and so were at a course level of granularity.

In the Finley charts of the DAES reports,

- The knowledge gained from the Finley charts is that the dependencies are of some programmatic importance—they can affect the course of the subject program—otherwise they wouldn't be mentioned. So of all the potential types of interdependencies that could exist among programs, the Finley charts show those that present a potential risk to the program in question (subject to the limitations of the format and the awareness of the program manager). The dependencies described by the Finley charts generally relate to some component or subsystem in the subject program (or system) that must be provided by, or is somehow dependent upon, the external program (or system). In many cases (actually, most cases) the external entity is a non-ACAT 1D program. There is no requirement for those programs to report their data to OSD via the SAR and DAES. In fact, the data for those programs will be held by the program office or their Program Executive Offices within the military department. This makes getting detailed data about the external program difficult.
- Also, the challenge with the Finley charts is that the nature of the dependency is usually not defined: It could be funding, schedule, or some technical issue. Given the shortcomings of the Finley charts as a way to represent programmatic interdependencies, other more objective representations of system interdependencies have been explored, particularly artifacts that describe the interconnections between the system in question and external systems. These data are in the Information Support Plan (ISP) that each major program generates as part of its milestone approval documentation. The difficulty with the ISP, however, is that the reports are more difficult to obtain, and recent changes in policy have made the data less analytically useful.

The data acquisition challenges could be summarized as follows: Although there are some allusions to the idea that various quantities presented in the reports are quantitatively obtainable through formulas or calculations, there is not much explanation as to how this is actually done or what the numerical values/ranges would be and whether these definitions are consistent across all programs. This information is crucial to building a state space model for the MDAP network. Also, the strategy for determining interdependencies seems to be a difficult. Also, given the time lag (DAES reports are generated monthly) and the level of data captured, often there was not variation in the data from one month to the next.

Analyzing Contract Data

We then deliberated on whether contract data would probably be a better data set for the type of time series based risk analysis we were considering. Instead of focusing on metrics related to contract value (looking for indicators of cost growth), we would instead look at the frequency of contract transactions.

The idea is that when a program is running smoothly, there's probably a baseline rate of contract modifications in the normal course of business (i.e., as funding is added, tasks are completed, deliverables are received, etc.). However, when something traumatic happens, like a test failure or other technical difficulties, we could probably expect significant contractual “churn” as previously-planned efforts are realigned to address the mission-critical issue.

The following is a possible scenario where the “churn” metric might be a more reliable indicator of program distress than cost: Consider a program that is composed of multiple components, each being developed under separate contracts (e.g., a satellite and its ground control segment). If, for example, the satellite has a problem in development (i.e., a test failure), the satellite contract will probably experience cost growth, but the ground control segment might actually experience a decrease in expenditures, as it has to slow down to accommodate delays in the satellite. So whereas costs might increase on one contract, they might be somewhat offset by temporary decreases in the other, which would muddy the “signal” seen at the overall program level. However, each contract would probably have to be re-scoped in order to increase the level of effort for the satellite and reduce the level of effort for the ground segment. Thus, both will incur additional contract “churn” as a result, which should be observable by plotting the frequency of contract modifications over time.

Figures 9, 10, and 11 are the time series of the contract “churn” for the three MDAPs. Each contract transaction reported in the Federal Procurement Data System–Next Generation (FPDS-NG) has an “issue date” indicating when the contract modification was signed. We plotted the frequency of contract actions over time.

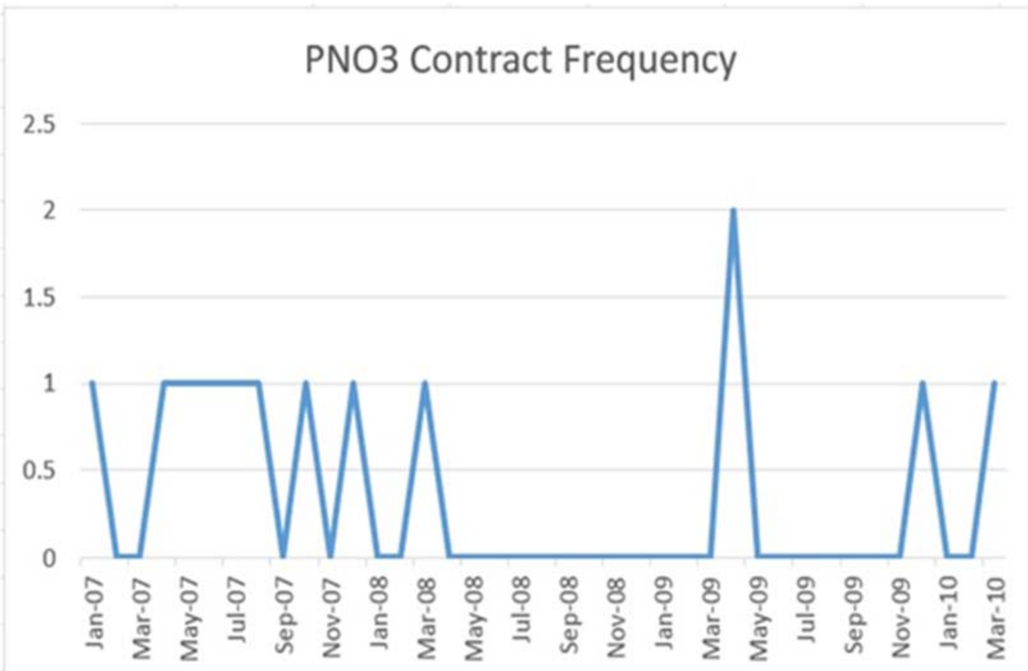


Figure 9. Time Series Data of PNO3-Related Issue Dates

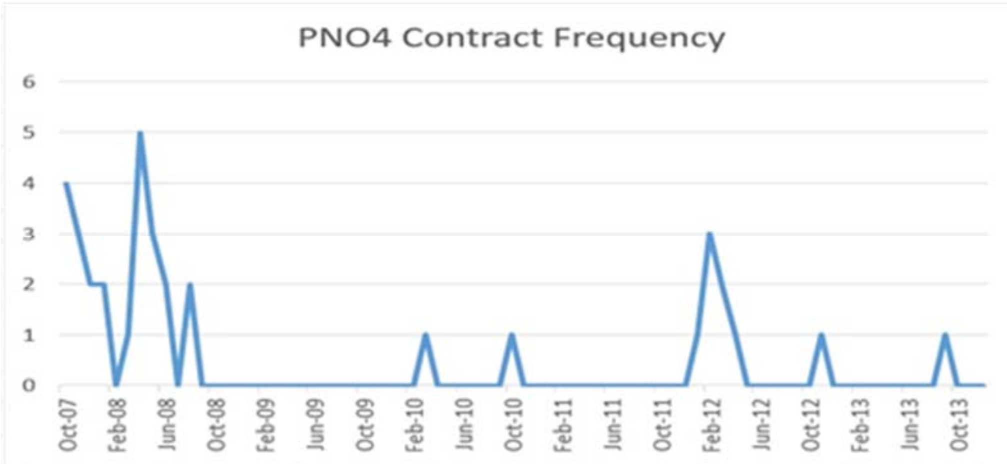


Figure 10. Time Series Data of PNO5-Related Issue Dates

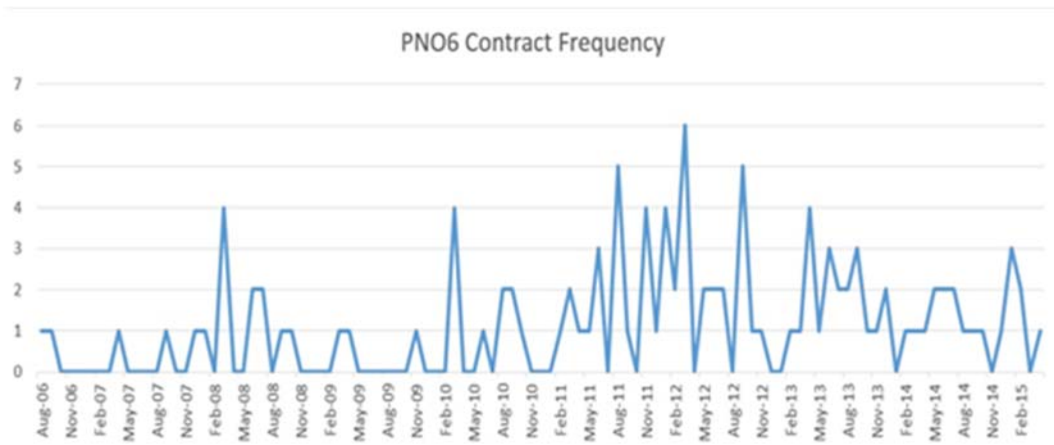


Figure 11. Time Series Data of PNO6-Related Issue Dates

In an effort to determine whether there is any type of correlation between the onset of significant contract churn in Figure 11 and program performance, we examined the breaches reported in the annual SARS data for PNO6. The December 2004, 2005, 2006, and 2007 SAR files show no APB or Nunn-McCurdy breaches, although the notes in the 2005 Threshold breach section state that there was a cost deviation from the key decision point-b approved APB even though there was no change in the total program cost as a result of the action. The 2009, 2010, and 2011 SARS show Schedule and Cost RDT&E APB breaches with varying levels of explanations. The December 2012 SARS indicates no such breach. We are continuing to study the executive summaries as well as SARS of future years in more detail.

Our observation from this examination of churn in contract data is that it does indeed have the volatility that could support the network modeling process. In addition to studying the PNO6 SARS data in greater detail as mentioned above, we are also trying to find contract data over a sufficiently long time scale to support our modeling analysis.

Conclusions and Future Work

In this paper, we have discussed our progress in our ongoing efforts to (1) study the impact of network topological characteristics on risk propagation and our methodology to quantify it, (2) evaluate the critical importance of quantifiable state features in order to assess network dynamics, and (3) describe our investigation into time-series data that could facilitate our analysis.

Our initial results on PRA analysis for a case study and the contract data time series are encouraging, and we plan to further investigate the scale-up of the PRA analysis as well as using the contract data towards building the network model.

References

Albert, R., Jeong, H. & Barabasi, A. (2000). Error and attack tolerance of complex networks. *Nature, 406*, 378–382.

Al-Mannai, W., & Lewis, T. (2007). Minimizing network risk with application to critical infrastructure protection. *Journal of Information Warfare 6*(2), 52–68.

Brown, M. M. (2014). Acquisition risks in a world of joint capabilities: A study of interdependency complexity. In *Proceedings of the 11th Annual Acquisition Research Symposium* (pp. 109–128). Monterey, CA.

Lanford, O., & Mintchev, S. (2015). Stability of a family of traveling wave solutions in a feedforward chain of phase oscillators. *Nonlinearity, 28*, 237–261.

Lewis, T. G. (2009). *Network science: Theory and practice*. Hoboken, NJ: John Wiley & Sons.

Mintchev, S., & Young, L. (2009). Self-organization in predominantly feedforward oscillator chains. *Chaos, 19*(4), 043131.

Raja, A., Hasan, M. R., & Brown, M. M. (2012). Facilitating decision choices with cascading consequences in interdependent program networks. In *Proceedings of the Ninth Annual Acquisition Research Symposium* (pp. 197–220). Monterey, CA.

Raja, A., Hasan, M. R., Rajanna, S., & Salleb-Aoussi, A. (2013). Leveraging structural characteristics of interdependent networks to model non-linear cascading risks. In *Proceedings of the 10th Annual Acquisition Research Symposium* (pp. 293–318). Monterey, CA.

Raja, A., Hasan, M. R., Rajanna, S., & Salleb-Aoussi, A. (2014). A scalable approach to modeling risk in the MDAP network. In *Proceedings of the 10th Annual Acquisition Research Symposium* (pp. 293–318). Monterey, CA.

Szell, M., Lambiotte, R., & Thurner, S. (2010). Multi-relational organization of large-scale social networks in an online world. In *Proceedings of the National Academy of Sciences of the United States of America, 107* (pp. 13636–13641). San Mateo, CA: Morgan Kaufmann.

Acknowledgments

We thank Professor Stan Minchev for his contribution and insights on building a formal model of MDAPs to determine the dynamics of the networks and its effect on performance. We also thank Eui Seong Han for his assistance with the data organization and analysis.

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

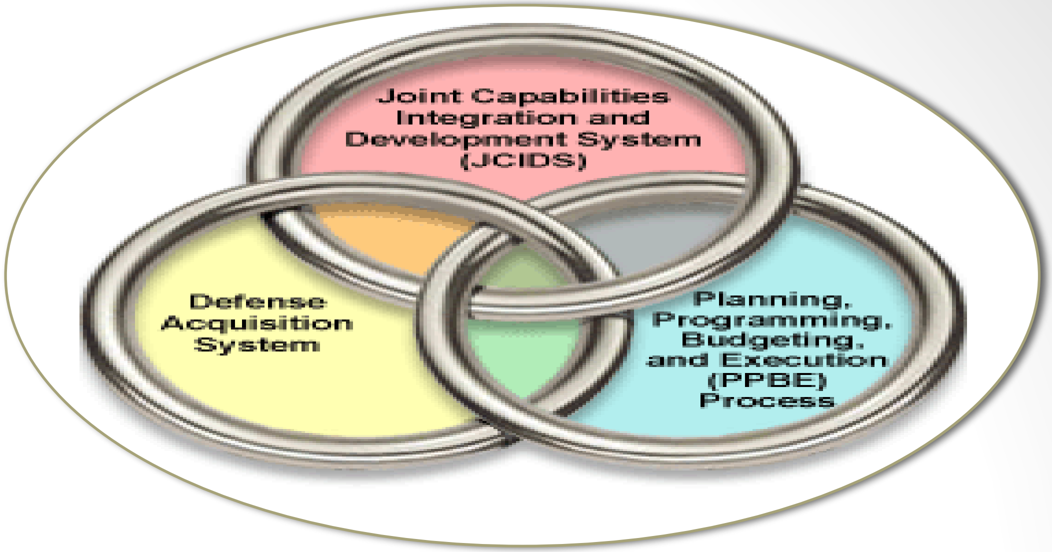
www.acquisitionresearch.net

Modeling Uncertainty and Its Implications in Complex Interdependent Networks

Anita Raja, Mohammad Hasan (UNL),
Rob Flowe (OSD), Brendan Fernes

Presenter: Anita Raja
Professor of Computer Science,
The Cooper Union, NY

Motivation



- Background:
  - Joint Capabilities.
  - Operating environment: uncertainty, complexity, rapid change and persistent conflict.
  - Integrated approach (WSARA 2009).
- Definition:
  - Cascading risk: Propagation of programmatic issues across networked programs due to the interdependency of one program upon the other.
- Research Question:
  - Study and quantify the impact of network characteristics on cascading risk.

Main Contributions

Hypothesis: The programmatic interdependencies between MDAPs have a profound influence on large-scale network performance over an extended period of time.

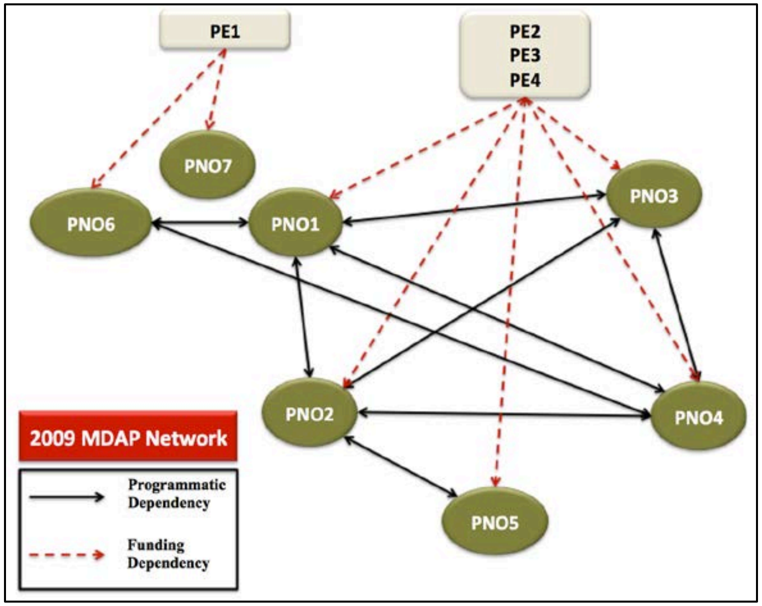
1. Define a metric to quantify influence of network characteristics on program performance.

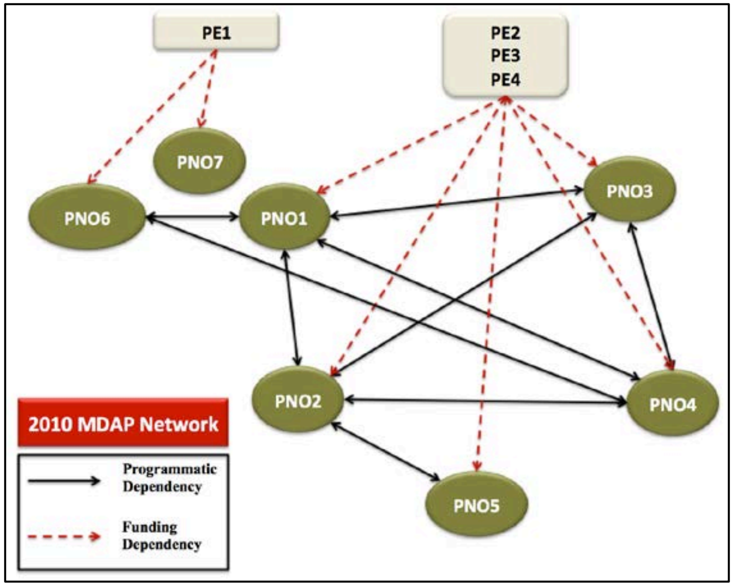
1. Determine if it is possible to formulate mathematical models that capture dynamics of complex networks and provide prescriptive actions.

Task 1: Risk Computation

- Probabilistic Risk Analysis (Lewis 2009) :
  - Compute risk to identify and manage critical nodes.
  - Why critical nodes?
  - Leverage network topology (extended PRA)
  - Extended risk for n-node network is $r_{ext}=\Sigma^{n}_{i=1}g_iV_iC_i$ where
    - g: Degree
    - V: Vulnerability
    - C: Consequence

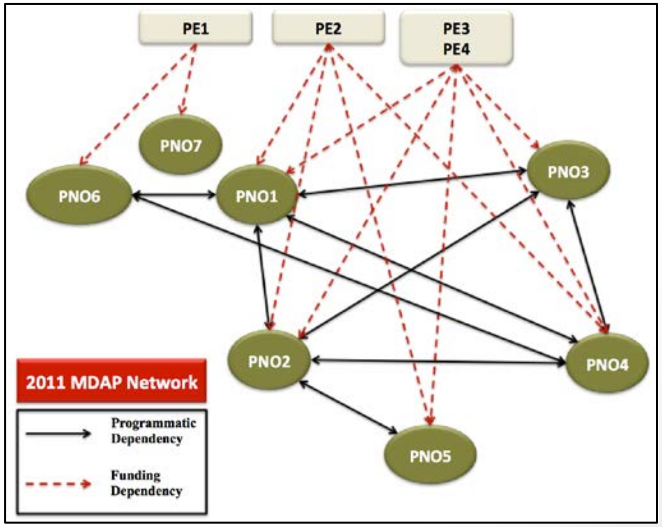
Case Study of a multiplex network



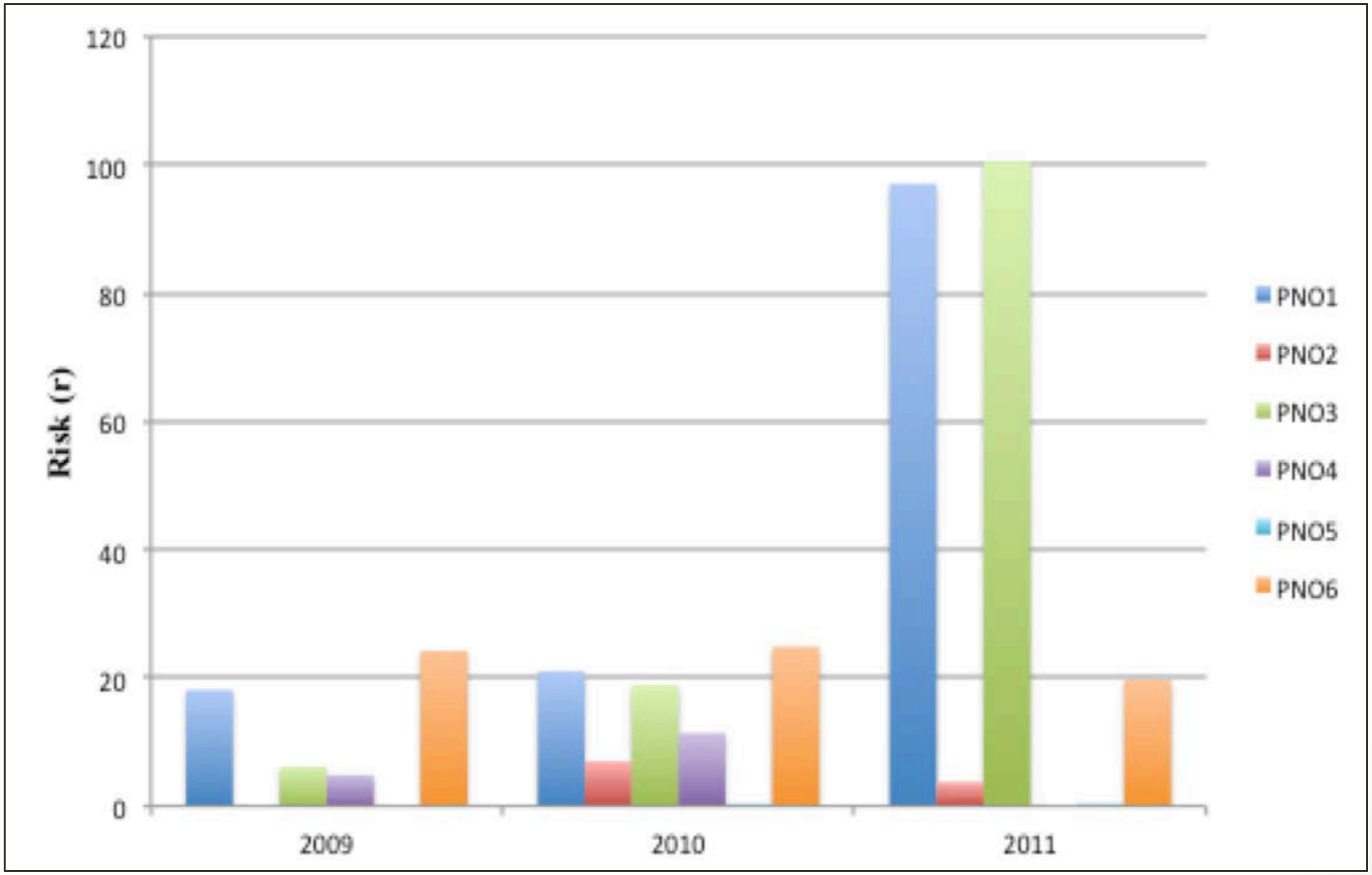


Critical Node Analysis for multiplex network

	Risk (r)		
	2009	**2010**	**2011**
PNO1	18.11	21.02	97.05
PNO2	0	6.97	3.75
PNO3	6.16	18.89	100.52
PNO4	4.88	11.39	0
PNO5	0	0.17	0.48
PNO6	24.22	24.9	19.7



Critical Node Analysis

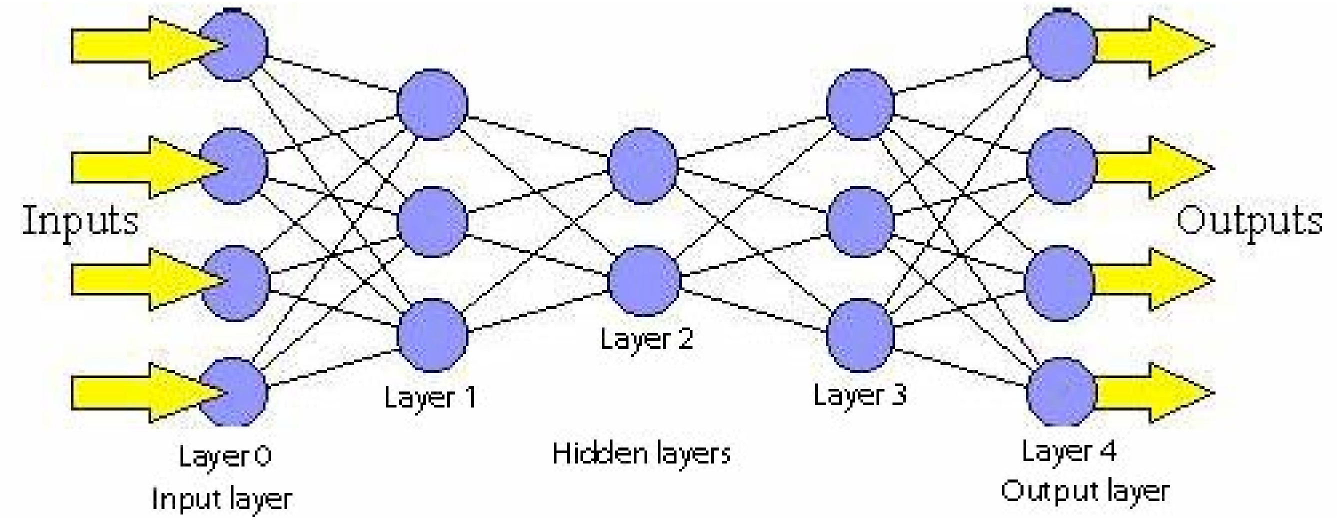


Task 1: Observations

- Network-centric approach helps capture exogenous effects on program performance.
  - Underscores importance of addressing multiplex network relationship among MDAPs.
  - Uses data from various reports (DAES, SAR, R Docs) to define multiplex network and vulnerability parameter for PRA analysis.
- Extended PRA technique
  - identifies critical (risky) programs in the multiplex network.
  - could be used to forecast a program's performance and avoid negative cascading effects.

Task 2: Feasibility Study

- Modeling interdependent networks as a coupled dynamical system and potentially adapting related algorithms.



- Does the system have any attractive equilibria?
- If good equilibria were discovered, an outcome could be to recommend a funding strategy that maintains equilibrium or guarantees a rapid convergence toward it.

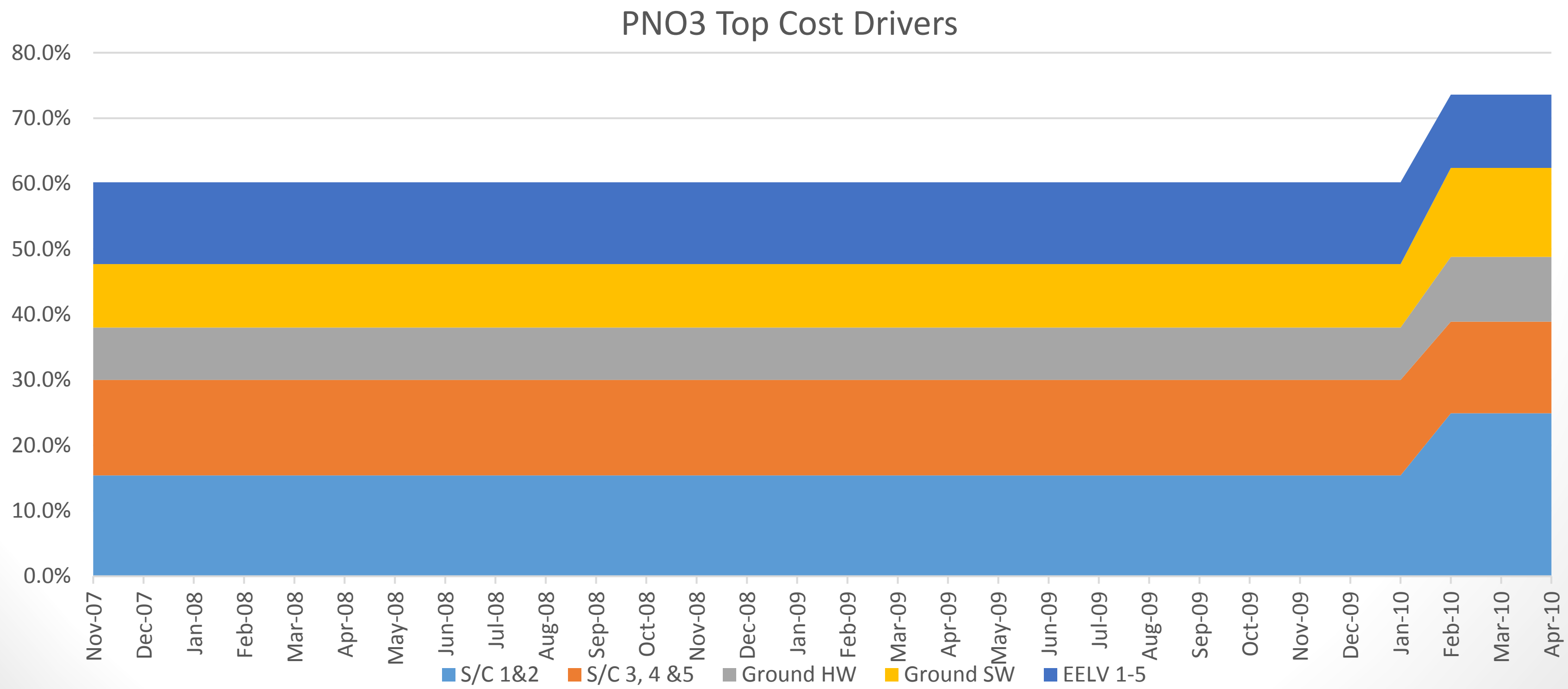
  - Ack: Prof. Stan Minchev, Eui Seong Han.

Methodology

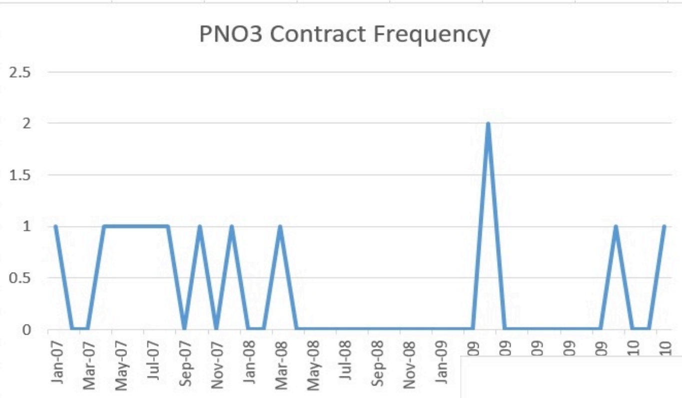
- Goal: Determine a network model
  - Centrality measure.
  - Strength of network connections including a precise form of the coupling formalism.
  - State features and action options (as in Raja 2012).
  - Reward optimization model capturing network dynamics.
- Process:
  - Determine variables that **evolve over time** and can be **measured numerically**, i.e., real numbers on a well-defined scale
  - Find time series of data on the variables chosen above.

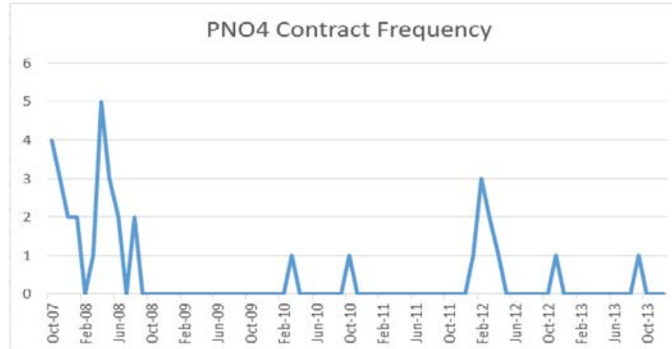
E.g.: Top Cost Driver

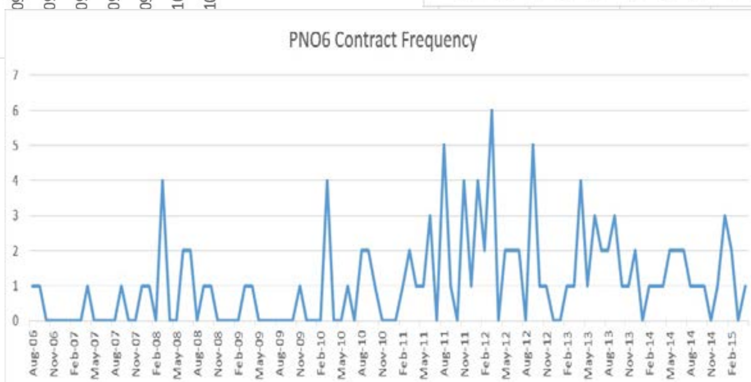
- Data source:
  - DAES reports of several MDAPS collected over a decade
  - Program Status, Top Cost Drivers, KPPs, Finley Charts.



Churn in Contract Data







- July 2008: new cost and schedule breach occurred per the PNO6 DAES data. We are investigating if spikes in the contract churn data predict a future breach.
- Missing data problem.
- 2009, 2010, 2011 PNO6 SARS show Schedule and Cost RDT&E APB breaches with varying levels of explanations.

Task 2 Observations

- Allusions that various DAES and SARS quantities are quantitatively obtainable through formulas or calculations. However unclear
    - how this is actually done;
    - what the numerical values/ranges would be;
    - whether these definitions are consistent across all programs.
- Given time lag (DAES reports are generated monthly) and the level of data captured, often there was not variation in the data from one month to the next.
- Churn data has the evolutionary characteristics that could facilitate network modeling process.

Future Work

- Risk computation: Further study the impact of network topology on risk propagation and our methodology to quantify it.
- Delve further into contract data and investigate effect on future performance wrt breaches.