ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2013) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

Conference on Systems Engineering Research (CSER'13)
Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.

Development Interdependency Modeling for System-of-Systems (SoS) using Bayesian Networks: SoS Management Strategy Planning

Seung Yeob Han and Daniel DeLaurentis

Purdue University, West Lafayette, IN 47907-2023, USA
(Email: simonhan, ddelaure@purdue.edu)

Abstract

Managing the development and evolution of a system-of-systems (SoS) capability remains a challenge due to, among other reasons, the complex interdependencies between participating systems. One form of complexity stems from the tendency of interdependencies to propagate between systems; disruptions in the development of one system may propagate to other dependent systems in successive cycles, creating schedule and cost overruns. Event tree methods and Bayesian Networks (BNs) are used in this paper to quantify development interdependencies between systems and assess cascading development risks. In addition the approach also allows inputs (e.g. development failure rates of participating systems) to be updated automatically for better decision-making. A primary output of the approach is the quantification of development interdependencies and the identification of critical systems with respect to propagating effect levels. This method when applied to a synthetic problem, as a proof-of-concept, demonstrates the robustness of the proposed approach in tackling risk interaction that arises from the cascading effects of development disruptions and clearly illustrates that the propagating effects depend not only on SoS architecture, but on development failure rates of participating systems as well. The outcomes of the analysis provide a support for decision makers to manage risk in development of a SoS with complex interdependencies.

Keywords: Development Interdependency; System-of-Systems (SoS); Bayesian Networks (BNs); Cascading Effects; Risk Management

1. Introduction

A system of systems (SoS) is a collection of distributed systems that operate independently but that also interact with one another to achieve a capability not achievable by individual systems alone (in contrast, in a monolithic system, hardware or software components are integrated to form a single entity). SoSs are found in various domains, such as aerospace, defense, and communication networks [1], [2] and are distinguished by their operational and managerial independence [3]. Whilst giving a SoS the characteristic capability of achieving its objectives, this interdependency can also cause failures to cascade through the SoS thereby causing potential development delays. Therefore, these properties of independence and interdependence make project management of SoSs challenging during the development process. For example, the Joint Strike Fighter (JSF, or F-35) continues to face schedule and cost overruns, partially due to the cascading effects of component systems failing to meet requirements. Many surveys show the schedule and cost overruns are serious problems in a SoS, especially defense programs. The Government Accountability Office (GAO) report estimated that the 98 Major Defense Acquisition Programs

(MDAPs) from FY2010 collectively overran their schedule by an average of 22 months and budgets by $402 billion [4]. According to an independent report by the British Ministry of Defense, defense programs overran their schedule by an average of 40% leading to overall cost increases by 40% [5]. Therefore, an adequate assessment of the cascading effects of risk among interdependent systems during the development process has the potential to reduce cost and schedule overruns. The rationale behind this is that such analyses can reduce risk of a project management.

Several approaches have been developed to analyze systems interdependencies during operations and associated propagating impacts in different domains [6], [7], [8]. These studies focus on analyzing operational interdependencies between systems using data that is more readily available than the data required for analyzing development dependencies. In such scenarios researchers are restricted to carrying out analysis with limited information such as sparse data or expert opinions, and are hence struggling to quantify development dependencies. Therefore new approaches are needed not only to quantify the development dependency strengths with cascading system failures in a SoS during the system development but also, to deal with uncertainty.

In this paper, we propose a method to quantify the development dependency strengths between system builders. The method employs Bayesian Networks (BNs) to represent interdependencies between system builders in a SoS capability development context. The BN uses two inputs: 1) development failure rates of system builders and 2) dependency strength between system builders. BNs are particularly suited to such problems given their efficacy for representing causal relationships between systems involving uncertainty. Uncertainty is represented in a BN using beta distributions which increases the robustness of model outcomes. The BN is used on a synthetic problem to compute the impact of interdependencies between system builders. Results are described in the context of their use by decision makers to manage risk in development of a SoS with complex interdependencies.

2. Development Interdependency Modeling for System-of-Systems (SoS)

There are two ways to develop a new SoS: 1) integrate only existing off-the-shelf systems or 2) deploy nascent and inchoate systems with existing ones. In many cases, the latter is selected for the sake of the advancement of SoS wide capability. It is important for project managers to effectively estimate possibilities of developing new systems. However, estimating development possibilities for all activities is a challenge due to the complex interdependencies between systems activities organized for a SoS capability. In this paper, we use failure rates of systems activities to represent possibility levels for a system development. Once a failure rate of each activity (in consideration with its interdependent activities) has been estimated it can be used for better decision making to reduce schedule and budget overruns.

We develop an integrated simulation model to estimate failure rates of system activities in a SoS. The model identifies the system which is most susceptible to disruption propagation impacts. The outputs from the integrated simulation model are the estimated failure rates of all activities at scheduled time.

The integrated simulation model is comprised of five principal steps (Fig. 1) as described by the following:

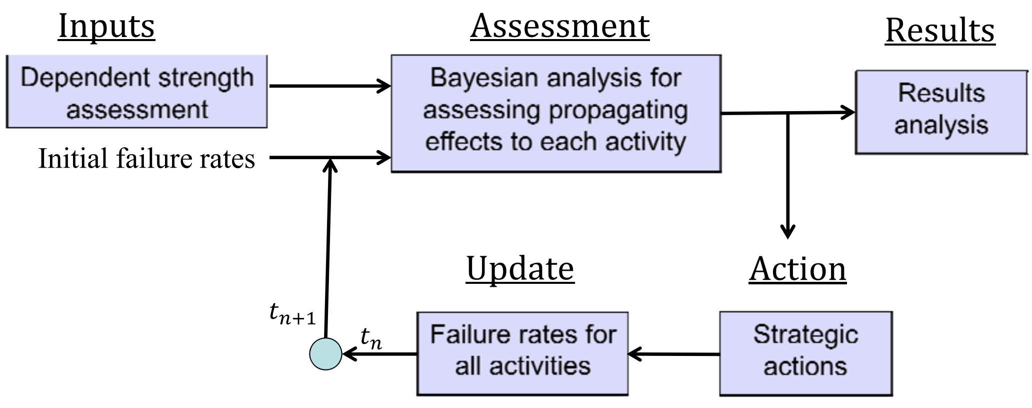


Fig. 1. Overview of the integrated simulation model for development interdependency analysis

2.1. Inputs – Initial development failure rates and dependency strength

The first step in the modeling development interdependency analysis is to estimate initial development failure rates and the dependency strength of activities. Initial development failure rates of system activities depend on technology maturity levels and the system builder's potential for the new system development. If a system builder has mature technologies or high potential to develop a new system, then initial development failure rate for that activity is low. It is important to account for uncertainty in the model and data in order to generate reliable results. Hence beta distributions are used to represent initial development failure rates to address uncertainties. Beta distributions are a family of continuous probability distributions defined on the interval between 0 and 1, parameterized by two positive shape parameters (α and β). There are several reasons for justifying the use of beta distributions in the proposed model. First, beta distribution allows for the representation of various types of probability information [9]. Fig. 2 presents various shapes of beta distributions according to two positive shape parameters. Second, when probability information is unavailable, the beta family is the best choice for determining the most conservative probability information [10].

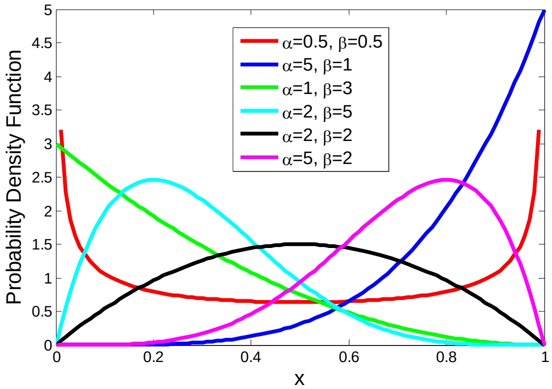


Fig. 2. Various shapes of beta distributions

The second major input, dependency strength, plays an important role in analysing propagating effects among interconnected systems. We use *requirement connectivity matrix* and *system maturity levels* to quantify dependency strength between system activities. In this paper, dependency refers to requirements dependency. We assume that the individual component systems in our SoS are built independently. If there does not occur any problems during system development, development of the component systems should not have propagation of disruption between each other. However, if a disruption of the system development happens, then this disruption may propagate. For example, Unmanned Aerial Vehicle (UAV) and a satellite are constituent systems of a SoS. One of the SoS requirements is to send/receive data with high bandwidth between UAV and the satellite. To achieve this requirement, system builders should communicate with each other to develop compatible data transmission systems. This dependency strength is called '*requirement dependency*'. The dependency strength represents the chance that disruption in the activities of the UAV builder will propagate to the Satellite builder. The detailed process to estimate requirement dependency strength is as follows.

Connectivity matrix for requirement dependency

The connectivity matrix is a column matrix indicating which requirements of a system depend on the other systems. Suppose in our last example, the UAV and the Satellite have three requirements each and one pair of requirements is dependent. If UAV's 3rd requirement is dependent on the Satellite's 1st requirement, then connectivity matrix for UAV and the Satellite can be represented by $Con_{UAV \rightarrow Sat}=[0\ 0\ 1]^T$ and $Con_{Sat \rightarrow UAV}=[1\ 0\ 0]^T$ respectively. The connectivity matrix only allows binary entries 0 or 1, where '0' represents independence and '1' represents independence.

Conditional probability of a requirement failure given a system builder failure

The second step is to quantify requirement dependency strength through estimating conditional probabilities of requirement failures using technology maturity scales. In the development process for high technology systems, system builders may not deliver new systems on time due to lack of technology maturity. In this paper, we use a system-focused prescriptive metric entitled System Readiness Level (SRL) to estimate the technology maturity scales of a requirement. A requirement can be hierarchically decomposed into several systems needed to achieve the requirement capability. Fig. 3 represents the hierarchical representation of requirements and the capability of an entire system.

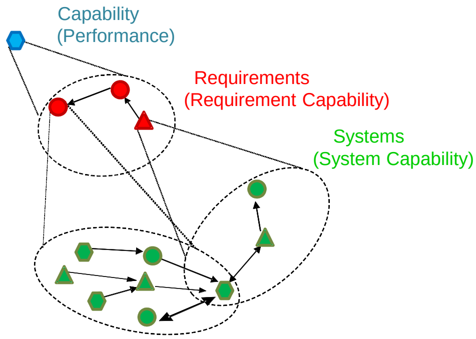


Fig. 3. Hierarchical representation of an entire system

SRL is defined as a function of Technology Readiness Levels (TRLs) of component systems and Integration Readiness Levels (IRLs) of the integrations. TRLs are a systematic metric/measurement, invented by NASA that supports assessments of the maturity of a particular technology and the consistent comparison of maturity between different types of technology [11]. IRLs provide an integration specific metric to determine the integration maturity between two or more component systems [12]. Requirements can be interpreted as systems with several component systems. Therefore we can estimate the technology maturity scales for requirements using SRL metrics. In order to address uncertainty into SRL metrics, Tan, et. al. [13] developed a probabilistic approach which provides a SRL probability distribution incorporated information of relative frequency of the TRL/IRL values provided by system evaluators.

SRL can take values between 0 and 1. A system with a higher SRL value represents a matured system. The matured system has a smaller residual of the SRL value, unity minus a SRL value. We use mean values of SRL probability distributions to calculate the conditional probability of a requirement failure by normalizing residuals of SRL values for all requirements. If three requirements of UAV have SRL values of 0.4, 0.8, and 0.5 respectively, the conditional probability of requirement 1 failure given the UAV builder failure is as follows.

$$P(UAV_{Req1}=F|UAV=F)=\frac{SRL\,residual\,of\,req_1}{\sum_{i=1}^{3} SRL\,residual\,of\,req_i}=\frac{(1-0.4)}{(1-0.4)+(1-0.8)+(1-0.5)}=0.46 \quad (1)$$

After applying this process to all requirements, we can obtain a matrix including conditional probabilities of all requirements, P(UAV_{Req}=Failure | UAV=Failure)=[0.46 0.15 0.39].

Quantify requirement dependency strengths

Requirement dependency strengths are obtained by multiplying the connectivity and conditional probability matrices. The dependency strength represents the probability that disruption of a system builder will propagate to its dependent system builder. The failure of UAV builder may propagate to the satellite builder with the probability of 0.39 as calculated in the following equation:

$$Dependency\,Strength_{(UAV\rightarrow Sat)} = P(UAV_{Req} = F\,|UAV = F)\times Con_{(UAV\rightarrow Sat)} = 0.39 \quad (2)$$

2.2. An Interdependency Analysis of a System-of-Systems using a Bayesian Network

A Bayesian Network

A Bayesian Network is a directed acyclic graph (DAG) whose nodes are random variables and whose edges directly influence one another. Local probabilities represent the nature of the dependence of each variable (node) on its parents. Probability information in a Bayesian Network model is defined through these local distributions. A node with no parent node in the Bayesian Network model denotes a random variable and its associated probability distribution. A node with its parent node(s) can be represented as a conditional probability distribution (CPD). The first important step to build the BN is to construct the network of interests while considering dependencies between nodes and to estimate failure rates of nodes. Requirement dependency strength and failure rates of system builders mentioned in Section 2.1 are used to form the BN.

In this paper, a Bayesian Network (BN) is adopted to analyze interdependencies between systems. The BN can graphically represent interactions among multiple components and provide the basic structure for analyzing and visualizing the development SoS model. The BN is a probabilistic tool that evaluates networks of systems with respect to disruption propagation in developing systems for a SoS. The evaluation not only identifies critical components from a risk perspective, but also can show disruption and dependency effects on total expected development time of the SoS capability.

Propagating system failures through interdependencies

There are two sources of system failure in a SoS context: inherent and propagating. If a system fails on its own, then it is called an inherent failure. However, if a system failure is caused by propagating effects from interdependent systems, it is then called a propagating failure. It is therefore important to fuse all failure information, inherent and propagated.

Fig. 4 shows a simple Bayesian Network where the node Y has N parent nodes. This paper focuses on binomial failures for a node. For example, each node in the Bayesian Network can only take values 0 or 1 to represent the status of the component as 'failure' or 'working', respectively. This is a limiting factor of the approach which we will revisit at the conclusion.

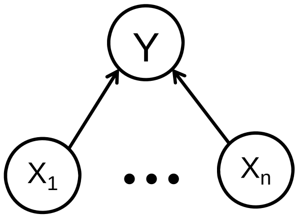


Fig. 4. A Bayesian Network representation

Consider that each node has its own inherent failure rate defined by a beta distribution and node Y has n parents, $X_1 \ldots X_n$. A Beta distribution is parameterized by two positive shape parameters, denoted by s_n+1 and s_n+n_n+1. These two positive parameters are interpreted as the number of failures and survivors respectively when s_n and n_n are integers [14]. Let **PA**(Y) denote the set of the parents of the node Y, i.e.$\{ X_1 \ldots X_n \}$. According to the law of total probability, the propagating failure rate of node Y is

$$p(Y=0)=\sum_k CPD_k \, p(PA(Y)=k) \tag{3}$$

where $CPD_k \equiv p(Y=0|PA(Y)=k)$, k is all combination of parent node values. For example, if **PA**(Y) includes two parent nodes (X_1 and X_2), then $k=\{\{0,0\},\{0,1\},\{1,0\},\{1,1\}\}$. Therefore, a node with two parent nodes has four CPD_k values. CPD_k values here indicate the dependency strength of a failure propagating to a dependent system. For instance, if node X_1 (or X_2) fails, then node Y has 30% (or 20%) of chance to fail by a propagating effect of the node X_1 (or X_2) failure. In this case, all CPD_k values are determined: $p(Y=0|X_1=1, X_2=0)=0.3$, $p(Y=0|X_1=0, X_2=1)=0.2$, $p(Y=0|X_1=0, X_2=0)=0$, and $p(Y=0|X_1=1, X_2=1)=0.5$. Analytical solution, $p(Y=0)$, for the propagating failure rate of node Y is very likely to be non-parametric due to its complicated functional form. For computational convenience, we use the approach in reference [15], [16] to approximate the propagating failure rate with a beta distribution having the same first two moments. Let beta(b,c) denote the beta distribution of the propagating failure rate of node Y. Then, the first two moments of this distribution are

$$M_1=E(Y)=\frac{b}{b+c}, \text{ and } M_2=E(Y^2)=\frac{b+1}{b+c+1}E(Y) \tag{4}$$

The first moment of p(Y=0) is the mean:

$$M_1=E(p(Y=0))=E(\sum_k CPD_k \, p(PA(Y)=k))=\sum_k CPD_k \prod_i E[p(PA_i(Y)=j)] \tag{5}$$

where j is the value for $PA_i(Y)$ in the set of k. For computational ease, Eq. (5) can be further written as the follows.

$$M_1=E(p(Y=0))=\sum_k CPD_k \prod_i \{ j_i - E[p(PA_i(Y)=0)] \} = \sum_k CPD_k \prod_i \left\{ j_i - \frac{s_i+1}{n_i+2} \right\} \tag{6}$$

The second moment of p(Y=0) is the mean of $p(Y=0)^2$:

$$M_2 = E(p(Y=0)^2) = \sum_k CPD_k \prod_i \left\{ j_i^2 - 2 j_i \frac{s_i+1}{n_i+2} + \frac{(s_i+1)(s_i+2)}{(n_i+2)(n_i+3)} \right\} \tag{7}$$

Finally, we can define two parameters, b and c, for a beta distribution of p(Y=0) as

$$b = \frac{M_1(M_1 - M_2)}{(M_2 - M_1^2)}, \quad c = \frac{(1-M_1)(M_1 - M_2)}{(M_2 - M_1^2)} \tag{8}$$

Now, node Y has two different beta distributions, one for inherent failure rate and one for propagating failure rates encapsulated in Eq. (8). These two beta distributions are integrated to get the new failure rate distribution of node Y. This task can be easily completed using the same process for obtaining fusion of all propagating failure information mentioned above, with different CPDs indicating 100% propagating effects. After applying these two fusion processes (the first being the fusion of propagating effects from dependent systems for the propagating failure rate and the second being the fusion of both inherent and propagating failure rates for the new integrated failure rate) to all nodes in a network, we can obtain the beta distributions of the new failure rate information including the propagating effects for all nodes. This result can be used to determine the critical (vulnerable) component and the expected development time for the SoS.

2.3. Update of Failure Rates for all activities

The failure rates of the system development may vary with time. If the system builder follows the schedule without any hiccups during the system development phase, development risk may decrease. In addition to evolution of the failure rates, uncertainty values in the failure rates of the systems decrease as time elapses. For example, assume that there are N times that the schedule is checked during a system development (Fig. 5). At each time step, a project manager estimates the failure rates of the system. In the beginning of the system development, uncertainty in the failure rates is high. However, once he knows the status of the system development at the time t_n, he may estimate more accurate failure rates of building the system. Therefore, for an accurate and reliable result, the proposed method includes analysis of update of failure rates for all activities with uncertainty.

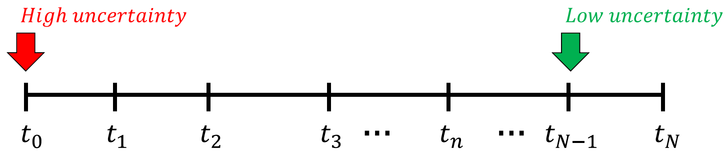


Fig. 5. Uncertainty on the time steps

In this paper, we use event tree analysis and beta distributions to update failure rates of all systems. An event tree is a commonly applied technique used for identifying the consequences following the occurrence of potential events such as failures or successes. It was first applied in risk assessments for the nuclear industry but is now utilized by other industries such as chemical processing, offshore oil and gas production, and transportation. We can estimate the outcome of the SoS development failure rate by quantifying the event tree diagram. Fig. 6 shows a very simple event tree structure for the development schedule of a SoS. All events are the scheduling checks during the SoS development. The branch points consider the failure and success of each event. The outcomes determined by the end point of each event tree branch identify a different value for failure or success following the initiating event. Total failure rate of a SoS development can be obtained by summing up all outcomes of failures at the end points of the event tree. If we know the initial failure rate and conditional probability of the system failure given the system failure or success at the previous step, then the event tree quantification is the simple task of multiplying the probabilities of passing along each branch point on any path through the diagram by the conditional probabilities. For example, let the initial system failure rate and conditional probabilities of the system failure probabilities given failure and success be $P(F_{t0})=0.1$, $P(F_{t1}|F_{t0})=0.4$, and $P(F_{t1}|S_{t0})=0.6$. Then system failure rate $P(F_{t1})$ at time t1 can be calculated using the following equation. All failure rates at any time steps can be estimated in the same way.

$$P(F_{t1}) = P(F_{t0}) \times P(F_{t1}|F_{t0}) + \{1 - P(F_{t0})\} \times P(F_{t1}|S_{t0}) = 0.1 \times 0.4 + 0.9 \times 0.6 = 0.56 \tag{9}$$

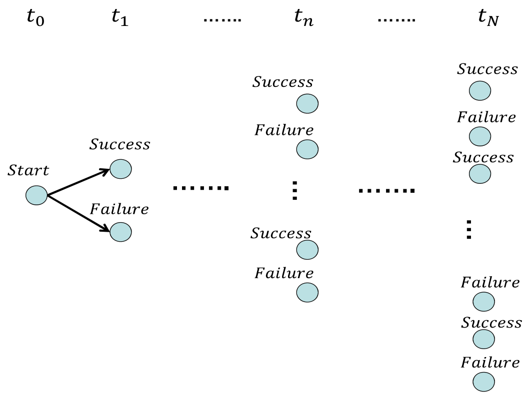
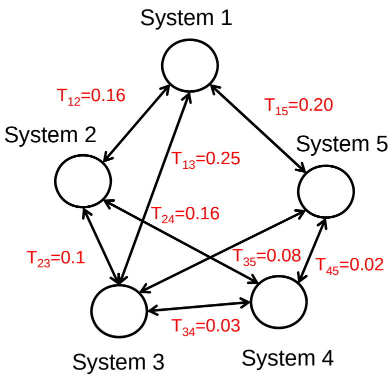


Fig. 6. A very simple event tree structure for an example development schedule

The historical data is used to estimate the initial failure rate and conditional probability of the system failure. In the absence of such information, we can also use uninformative distributions as initial inputs such as, uniform distribution. As time elapses, we observe the results at each time step. These observations can be used to update conditional probabilities of the system failure by adding weighted numbers to parameters of beta distribution. This process allows uncertainty of the system failure to decrease as time goes by.

3. Research Result – A Synthetic Demonstration Problem

A simple synthetic problem is formulated and solved to demonstrate the proposed approach. Fig. 7 shows the representation of a five-system network, where, the development of system 1, here denoted by S1, depends on the development of system 2, 3, and 5. This implies that a failure from one system development process affects the development of dependent systems due to requirement interdependencies between systems. For system 1, a failure cascades from system 2, 3, and 5 to system 1. The T values indicate the requirement dependency strength and correspond to the conditional probability of a failure propagating to a dependent system. For instance, if system 3 fails, then system 1 has 25% of chance to fail by a propagating effect of the system 4 failure. In order to estimate the requirement dependency strengths for all pair of systems, all systems should be decomposed into requirements and further into constituent systems. Then connectivity matrix and system readiness levels (SRLs) can be obtained using requirement relationships and TRL/IRL of constituent systems. Finally, the proposed method mentioned in section 2 allows us to estimate the requirement interdependency strengths. For simplicity, we skip over the detailed process entailed to obtain these values. The table in Fig. 7 summarizes initial failure rates for all systems and conditional distributions of failure rates in terms of beta distributions, and expected development time for each system.

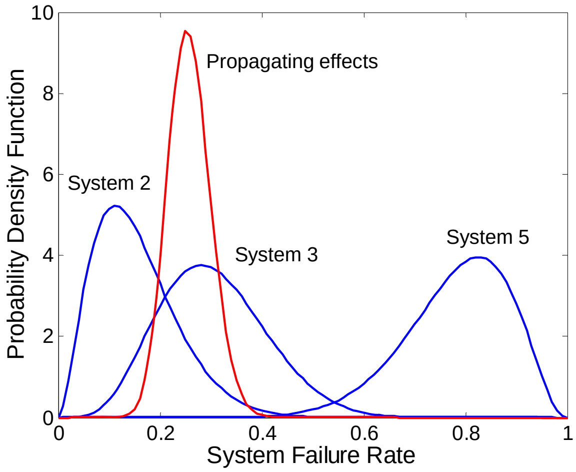


* F: Failure, S: Success, t_n: time n, t_{n-1}: time n-1

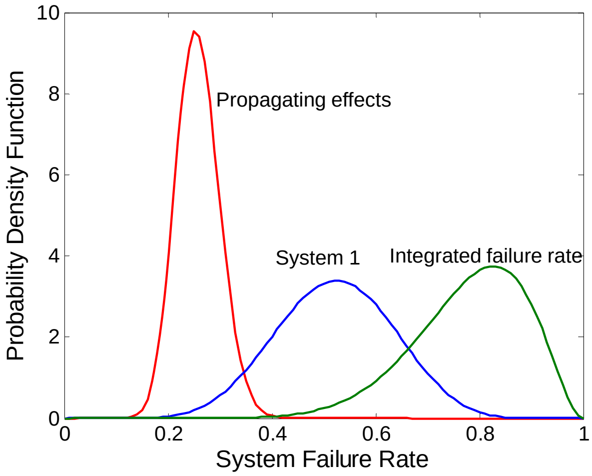
System	Failure rate distribution	$P(F_{tn}\|S_{tn-1})$	$P(F_{tn}\|F_{tn-1})$	Expected development time
System 1	Beta (10, 9)	Beta (3,8)	Beta (6,2)	6 (months)
System 2	Beta (3, 16)	Beta (1,10)	Beta (4,4)	8
System 3	Beta (6, 13)	Beta (3,8)	Beta (3,5)	5
System 4	Beta (9, 9)	Beta (4,7)	Beta (5,3)	9
System 5	Beta (12,3)	Beta (5,6)	Beta (7,1)	6

Fig. 7. A five system development network and all input information for the analysis

Consider the information fusion of failure rates of system 2, 3, and 5 with system 1 (Fig. 8). System 1 is connected to three dependent systems, 2, 3, and 5, with interdependency strength of 0.16, 0.25, and 0.20 respectively. Systems 2, 3, and 5 have their own inherent failure rates with beta distributions shown in Fig. 8a. The propagating failure rate on system 1 is easily calculated through the proposed approach, based on the given information about inherent failure rates and conditional probability for propagating effects. In Fig. 8, blue lines denote the inherent failure rate distributions for systems and red lines denote the propagating failure rate on system 1. The green line in Fig. 8b represents the integrated failure rate for system 1. The mean of the system 1 integrated failure rate represents an increase of 0.26 over its inherent rate value due to the propagating effects from dependent system failures. Fig. 9 shows the mean values of propagating effects for all systems. These values depend on the number of dependent systems and the failure rates of dependent systems. System 1 has the highest propagating effects, indicating strong dependencies with numerous other systems. It also has a higher probability to be disrupted by the failure of other systems during the development process. Since it is hard to control this kind of failure, the design team for system 1 should consider these propagating effects when scheduling the development time.



(a) Propagating failure rate



(b) Integrated failure rate

Fig. 8. Information fusion of failure rates of system 2, 3, and 5 with system 1

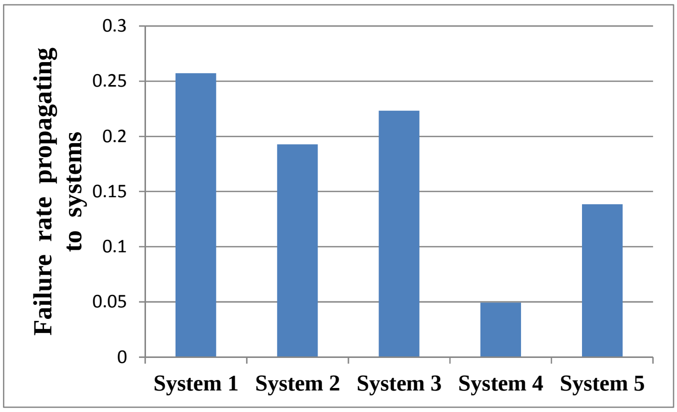


Fig. 9. Mean values for failure rates propagating to systems

The same synthetic problem is now used to measure the effects of disruptions and dependencies. The total expected development time is adopted to measure development time of a SoS capability. We assume that each system manager sets up the expected development time for their system as defined in the table in Fig. 7, indicating that in the absence of any failures, each system can be completed in the expected development time. Each system also has a development delay time due to its failure rate, calculated as the product of failure rate and the expected development time. For instance, if a system's failure rate is 0.6 and expected development time for the system is 1, then the project team of this system needs 1.6 times the normal duration to complete it. Therefore, the total expected development time can be formulated as the follows:

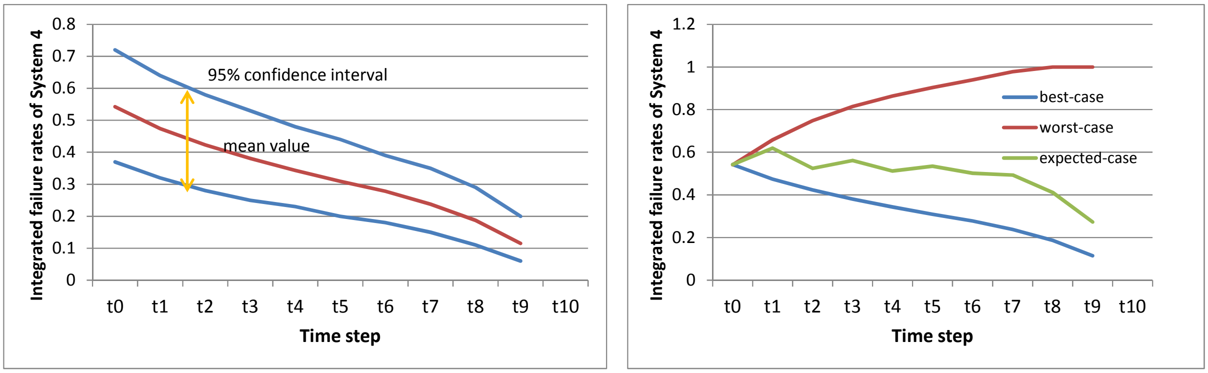
$$Total\ expected\ development\ time = \sum_i (1 + failure\ rate_i \times expected\ development\ time_i) \quad (10)$$

Table 1 shows the expected development time with/without considering disruption and dependency effects. The disruption and dependency effects increase total development time by 1.6 times. This schedule overrun usually ends up with cost overruns. Therefore when decision makers take a decision on a new SoS capability, disruption/dependency levels of required systems should be considered.

Table 1. Disruption and dependency effects to expected development time

Systems	Expected development time (months)		
	without disruption & dependency effects	with disruption effects & without dependency	with disruption & dependency effects
System 1	6	9.1	10.7
System 2	8	9.2	10.8
System 3	5	6.5	7.6
System 4	9	13.4	13.9
System 5	6	10.7	11.5
Total	34	48.9	54.5

Suppose there are 10 time steps for scheduling checks during the development process. In the beginning of the SoS development, it it hard for decision makers to estimate whether this project will end on time or not due to the high uncertainty. However, as time goes by, decision makers gain confidence in the results. Fig. 10 (a) shows this pattern using the mean values of the integrated failure rate of system 4 with 95% confidence interval. In this case, we use the best case scenario to run the simulation, indicating that all schedules at every time step are met on time. These observations are also applied to update failure rate of system 4. Fig. 10 (b) represents three different cases - best, worst, and expected, of integrated failure rates of system 4. All observations in the worst case are failures to achieve the schedule at each time step. We also draw numbers from beta distribution of integrated failure rate at the previous step to define the expected case. System 4 can lie anywhere in between best and worst cases. If the integrated failure rates of system 4 shows an increasing pattern then decision makers need to devote more attention to system 4 or substitute it with an alternative system.



(a) (b)

Fig. 10. (a) Mean values of integrated failure rate of system 4 with 95% confidence interval, (b) Mean values of integrated failure rate of system 4 in three different cases

4. Conclusion

The development process of a system-of-systems capability is often affected by risks arising from interdependencies between constituent systems during the development life cycle. This paper adopts a Bayesian Network approach to analyze interdependencies using constituent system failure rates and requirement dependency strengths between systems. Propagating failure rates are calculated to describe interdependencies, with inherent failure rates being evaluated for individual systems. By the integration of these two failure rates, both currently expressed as beta distributions, a new failure rate distribution is achieved that holistically represents the true risk and accurately determines the critical components. Uncertainty represented in a Bayesian Network using beta distributions allows for enhanced robustness of model outcomes. Event trees are used to show evolution of constituent system failure rates during the development process. Results that consider both evolution of constituent system failure rates and propagating effects can help to manage risk in development of a SoS with complex interdependencies.

A simple, synthetic five-system SoS problem demonstrates the proposed framework. Results show the increase in integrated failure rate due to the propagating effects of interdependencies. The comparison of integrated failure rates among all systems is useful in identifying the most critical system in terms of which system generates the most vulnerability to propagating effects from dependent systems.

The specific Bayesian Network formulation approach in this paper rests on two basic assumptions. First, systems can only have one of two discrete states, such as ‘working’ or ‘failure’; thus continuous variables such as development percentage cannot be expressed directly. Second, the interdependency strengths between systems are assumed constant; thus there is no evolution of the interdependency strengths. Future work will address the relaxation of these assumptions. Furthermore, there is a need to develop a strategic action tool, as shown in the overview of the integrated simulation model in Fig. 1. This tool will allow decision makers to take better decisions in a SoS development process.

Acknowledgements

The authors acknowledge the sponsorship of this research by the Systems Engineering Research Center (SERC) under contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. In particular, the input and coordination of project Technical Monitor, Judith Dahmann and Scott Lucero, were of tremendous value to our research team.

References

1. W. A. Crossley, "System of Systems: An Introduction of Purdue University Schools of Engineering's Signature Area," presented at the Engineering Systems Symposium, MIT Engineering Systems Division, Cambridge, 2004.
2. V. Kotov, "Systems-of-Systems as Communicating Structures," Hewlett Packard Computer Systems Laboratory Paper HPL-97-124, pp. 1–15, 1997.
3. M. W. Maier, "Architecting Principles for Systems-of-Systems," Systems Engineering, vol. 1, p. 8, 1998.
4. G. L. Dodaro, “Defense Acquisitions: Assessment of Selected Weapons Programs,” Government Accountability Office, U.S. Government, Washington, D.C., 2010.
5. B. Gray, “Review of Acquisition for the Secretary of State for Defence,” Ministry of Defence, British Commonwealth, London, EN., 2009.
6. N. Xu and G. Donohue, K. B. Laskey and C. H. Chen, "Estimation of Delay Propagation in Aviation System using Bayesian Network," 6th USAEUROPE ATM Seminar, Baltimore, MD, June 2005.
7. S. Han and D. DeLaurentis, "Air Traffic Demand Forecast at a Commercial Airport using Bayesian Networks," AIAA Paper 2011-6905, Sep. 2011.
8. K. Nishijima, M. Maes, J. Goyet and M. H. Faber, "Optimal Reliability of Components of Complex Systems using Hierarchical System Models," special workshop on risk acceptance and risk communication, Stanford University, March 26-27, 2007.
9. C. S. Reese, V. E. Johnson, M. Hama, and A. Wilson, “A Hierarchical Model for the Reliability of an Anti-aircraft Missile System,” UT MD Anderson Cancer Center Department of Biostatistics Working Paper Series, 2005.
10. D. Dyer and P. Chiou, "An information-theoretic approach to incorporating prior information in binomial sampling," Communications in Statistics - Theory and Methods, vol. 13, pp. 2051 - 2083, 1984.
11. J. C. Mankins, “Technology Readiness Levels: A white paper,” Advance Concepts Office, Office of Space Access and Technology: NASA, 1995.
12. R. Gove, B. Sauser, and J. Ramirez-Marquez, “Integration Maturity Metrics: Development of an Integration Readiness Level,” SSE_S&EM_004_2007, In Stevens Institute of Technology, School of Systems and Enterprises, Hoboken, NJ, 2007.
13. W. Tan, J. Ramirez-Marquez, and B. Sauser, “A probabilistic approach to system maturity assessment,” Systems Engineering Vol 14 No. 3 2011 wiley periodicals, Inc.
14. M. D. Springer and W. E. Thompson, “The Distribution of Products of Independent Random Variables,” SIAM Journal on Applied Mathematics, vol. 14, pp. 511-526, 1966.
15. J. Liu, J. Li, and B. U. Kim, “Bayesian reliability modeling of multi-level system with interdependent subsystems and components,” IEEE International Conference on Intelligence and Security Informatics (pp. 252-257), Beijing, China, 2011.
16. W. E. Thompson and R. D. Haynes, “On the reliability, availability and bayes confidence intervals for multi component systems,” Naval Research Logistics Quarterly, vol. 27, pp. 345-358, 1980.
17. S. Han and D. DeLaurentis, “Acquisition Management for System-of-Systems: Requirement Evolution and Acquisition Strategy Planning,” 9th Naval Postgraduate School Symposium. Monterey, California, 2012.