Dynamic Complexity Measures for Use in Complexity-Based System Design

Jonathan Fischi, Roshanak Nilchiani, and Jon Wade

Abstract—**Difficulty predicting system behaviors introduces a certain level of complexity to a system design. The INCOSE systems engineering handbook indicates that system complexity is one of the seven key challenges influencing development when engineering a system of systems. The scope of this paper is first to survey systems engineering relevant definitions of complexity for latter application to the complexity evaluation framework. The literature search also includes state-of-the-art works on system complexity measurement. Before proposing new techniques, the current complexity-based system, and interface measurement and design techniques are explored. As the state-of-the-art only includes static/structural complexity quantification, entropy-based measures for dynamic complexity quantification are proposed. A sample system is evaluated using the proposed dynamic complexity measures and the results are discussed. The methods proposed herein provide a first step in the path to an enhanced system/interface complexity evaluation framework using dynamic complexity measures.**

Index Terms—**Complexity theory, IEEE systems journal, measurement techniques, system-level design, systems engineering and theory.**

I. Introduction

HISTORICALLY, systems have suffered failures due to complexity. Whether creating large systems of systems or smaller systems with a few subsystems, the discussion of system complexity is still largely subjective and considered an art. The systems engineering community needs the ability to define the attributes of complexity that manifest in systems and determine how they can be quantified. The ability to better understand complexity during the design process will reduce the likelihood of deploying fragile systems. This paper proposes dynamic complexity measures, which are a crucial piece of a larger complexity-based design framework.

This paper starts with a literature review, including various definitions of complexity, methods for measuring complexity and methods for performing complexity-based systems engineering. Subsequently, a reference definition of complexity is determined and a novel method for measuring dynamic complexity is proposed based on the reference definition. An entropy model is used to generate the metrics and a method for combining the metrics is proposed. Finally, sample systems are evaluated using the techniques.

Manuscript received July 23, 2014; revised November 24, 2014, February 25, 2015, and June 10, 2015; accepted August 12, 2015. Date of publication September 24, 2015; date of current version November 22, 2017.

J. Fischi is a Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: jonathan.fischi@stevens.edu).

R. Nilchiani and J. Wade are with the Department of Systems Engineering, Stevens Institute of Technology, Hoboken, NJ 07030-5991 USA (e-mail: roshanak.nilchiani@stevens.edu; dr.jon.wade@gmail.com).

Digital Object Identifier 10.1109/JSYST.2015.2468601

TABLE I
COMPLEXITY DEFINITION CORRELATION: ERDI [4]

Reference Definition	Erdi Complexity Aspects
Complexity determined by: **The system being observed**	System properties that allow for difficult to explain behavior (Circular causality, Feedback loops, logical paradoxes, strange loop)
Complexity determined by: **The capabilities of the observer**	System produces responses of scale unexpected by observer (Small changes resulting in out-of-proportion results)
Complexity determined by: **The behavior that the observer is attempting to predict**	System has some aspect that is unpredictable/ emergent

II. Complexity in Engineered Systems: A Literature Review

A. Definition of Complexity

The reference definition of complexity used in this paper is the degree of difficulty in accurately predicting future behavior such that complexity is determined by three critical elements, 1) the system being observed; 2) the capabilities of the observer; and 3) the behavior that the observer is attempting to predict. This definition is based on the definition of complexity from H. G. Sillitto and updated by J. P. Wade to include the dependence on the particular behavior that one is attempting to predict [1], [2]. Thus, a system can have both low and high complexity based on the behavior of interest. Similarly, Dodder and Dare of MIT indicate that Complex Adaptive Systems, a category of complex systems, have a high degree of difficulty in accurately predicting future behavior [3].

Peter Erdi's complexity definition is compared with the reference definition in Table I. Table II contains a comparison of the reference definition of complexity and the Sargut and McGrath definitions of complex systems. Sargut and McGrath indicate that predictability is the primary distinction between a complicated system and a complex system.

The MIT ESD list of complex system aspects (see Table III) appears to contain the reference definition. The MIT definition adds some specific system characteristics like dynamic behavior, quantity of components/interconnects, and entity interaction. When evaluating risks created in later sections it is useful to review the definition correlation tables here to ensure the relevant complexity aspects are fully explored.

TABLE II
COMPLEXITY DEFINITION CORRELATION: SARGUT AND MCGRATH [5]

Reference Definition	Sargut and McGrath Complexity Aspects
Complexity determined by: **The system being observed**	Unintended consequences result from simple actions
Complexity determined by: **The capabilities of the observer**	Observer has difficulty "making sense of a situation." "Human beings' cognitive limits mean that no manager (observer) can understand all aspects of the business (system)"
Complexity determined by: **The behavior that the observer is attempting to predict**	Predictability is the primary distinction between and a complex system and a merely complicated system. "Rare events are more significant than average ones." Interaction of various parts of the system result in unpredicted results

TABLE III
COMPLEXITY DEFINITION CORRELATION: MIT ESD [6]

Reference Definition	MIT ESD Complexity Aspects
Complexity determined by: **The system being observed**	System interdependencies may unintended; dynamic system; many intricately connected components
Complexity determined by: **The capabilities of the observer**	Components and interconnections, interactions, or interdependencies the observer finds difficult to describe, understand, predict, manage, design, or change
Complexity determined by: **The behavior that the observer is attempting to predict**	The complexity of a system can be quantifiable, but what makes a system appear complicated to the observer is subjective; How complicated the system appears depends on the nature of the interface of the system that is presented to its users.

B. Measuring Complexity

A source of complexity spectra is provided by Sheard and Mostashari [7]. The authors produced an overview of complex systems engineering-related literature. They proposed a complexity framework that contains most of the types of complexity they discovered in prior works. The framework was created based on the types of complexity determined. The subtypes are size, connectivity, and architecture which are related to structural complexity. Dynamic complexity is decomposed into short- and long-term aspects. The socio-political complexity is an effect of the development process and the inherent nature of the environment. A number of metrics were proposed for each category. The authors indicated limitations in the areas of theoretical/mathematical complexity, which likely correlates to conceptual complexity in software, causes of emergence, environmental complexity, and other areas. MITRE created a Systems Engineering Profiler tool to quantify program complexity (oriented toward the programmatic aspects) [8]. Shiner and Davison proposed an entropy-based measure of complexity [9]. The measure is based on the Boltzmann-Gibbs-Shannon entropy and the maximum entropy. The advantages over prior measures are the simplicity of computation and the nonextensive nature of the measurement. Yamano proposed a statistical complexity measure that he also claims to be computationally simple and nonextensive [10].

C. Complexity-Based Systems Engineering Design: State of the Art

Several research groups participated in the 2011 DARPA META II Complex Systems Design and Analysis (CODA) program. One of the relevant questions was regarding complexity metrics, the metrics relevance to schedule/cost/reliability, and how useful the metrics are in comparing alternate designs. A participating research group from Boeing Company, Purdue University, and Arizona State University selected a wide range of complexity metrics for evaluation against schedule/cost/reliability data sets from real world programs to create a composite, abstract complexity metric [11]. The team performed a statistical analysis to identify correlations between computed metrics and the observed behavior of cost/schedule/reliability and found strong correlations for particular metrics. The team indicated that comparing complexity metrics between system types would not be advisable. A participating research group from Massachusetts Institute of Technology developed entropy-based metrics and used them to characterize uncertainty in the system development processes [12]. The objective was to quantify complexity in a way that it can be managed during development. This research proposes a statistical method for generating a complexity curve based on entropy and maximum entropy. This method does not preclude dynamic complexity measurement, but neither does it present a unified/quantified model for ensuring it is captured. Another participating research group from United Technologies Corporation CODA produced an architectural design space exploration method, which includes complexity metrics. User defined data inputs and iterations on the model determine the optimal architecture, in which complexity metrics served as one of the inputs [13]. The model is based on static complexity metrics. While dynamic complexity was surveyed and metrics discussed, a unified/quantified model is not presented.

The Aerospace Corporation developed complexity analysis for NASA missions, which includes complexity index computations for comparisons between programs [14]. The index consists of system design parameters, the estimated complexity of the parameters, and an aggregate complexity for the system. Suh proposed axiomatic design principles to aid in design decision making, particularly for complex systems. Suh proposed a means to tie complexity to the probability of satisfying requirements [15].

The DARPA-funded Abstraction-Based Complexity Management effort completed by United Technologies Corporation proposes an abstraction based enumeration technique for complexity-based system design. Complexity numbers are created for potential system architectures which are then modeled against design and uncertainty models for determination of the optimal architecture [16]. The proposed complexity metrics are static.

A general principle of systems engineering is to group highly coupled things into subsystems. The subsystems in a system should then be as loosely coupled as possible in their interfaces with each other. The same principle is desirable when connecting systems. Similarly, software engineers desire low coupling and high cohesion for readability/maintainability and have developed software quality measurements for this purpose [17].

Several of the discussed techniques include coupling metrics as a part of their modeling.

III. Proposed Dynamic Complexity Metric

The metrics described in the state-of-the-art literature in Section II-C depend on structural interface complexity metrics. Here, we propose dynamic complexity metrics based on established complexity definitions.

A. Sufficiency of Complexity Definition

The reference definition of complexity suggested by the literature search is "the degree of difficulty in accurately predicting future behavior" such that complexity is determined by the system being observed, the capabilities of the observer and the behavior that the observer is attempting to predict. This reference definition will be used for subsequent complexity-based risk development in this effort.

When evaluating dynamic complexity metrics in Section III-B it is useful to review the definition correlation tables in the prior sections to ensure the relevant complexity aspects are fully explored. The reference definition is consistent with the other field-leading definitions of complexity and is used for subsequent metric development.

B. Metric Development

The Suh Information Axiom indicates that for a system with n functional requirements (uncoupled design), that information content is

$$\mathbf{I} = \sum_{i=1}^{n} \left(-\log(p_i)\right) \tag{1}$$

where

p_i Pr{parameter "i" satisfies functional requirement "i"}.
log() $\log_2()$ (with unit of bits) or $\log_e()$ (with unit of nats).

According to Suh, the minimum I is the optimal design since it has the least information and, consequently, the least complexity. Similarly, a quantitative measure for complexity is I if complexity and information are, in fact, correlated. Suh also indicated that a design is complex when probability of *success* is low, where the present research asserts that a design is complex when its future behaviors are difficult to predict [15]. This research adapts the Information Axiom using the reference definition of complexity. In our adaptation, having easily predictable behavior is less complex than not having easily predictable behavior.

The probability behavioral metrics are derived using the well-established Goal, Question, Metric Approach (GQM) method [18]. The principle of GQM is that project goals (the conceptual level) must drive questions (the operational level), which are represented by metrics (the quantitative level) (see Fig. 1).

In order to understand the limits of cognition/computation with respect to system behaviors, the goal in Table IV is set to "know and predict system behaviors." The capabilities of the observer will be evaluated as error bars to the metrics in a later stage. Timeframe t is adaptable to the set relevant for the system under evaluation. For example, a development system with a 20-year life cycle may be represented for $t =$ {end of development (1–2 years), short-term deployment (3–5 years), long-term deployment (6–20 years)}.

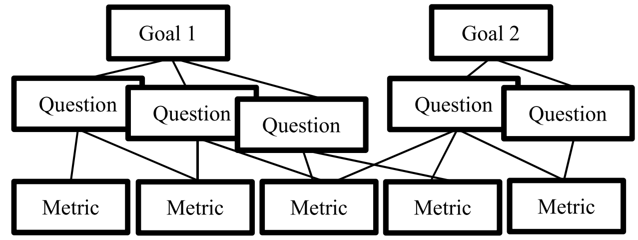

Fig. 1. GQM hierarchical structure [19].

TABLE IV
Generic Dynamic Complexity Metric GQM Model

Behavior Entropy Model	
Goal	Know and predict system behaviors
Question	What are the predicted system behaviors?
Entropy Metrics	**p1**= What is the probability that the selected systems behaviors for the actual mission cannot be easily/accurately predicted/navigated in timeframe t? **p2**= What is the probability that the selected systems behaviors for the actual environment cannot be easily/accurately predicted/navigated in timeframe t?

For the GQM Model, the following definitions will apply.

1) System Mission: includes requirements, uses, scale, scope.
2) System Environment: includes physical, programmatic, acquisition, political, stakeholder.

The Suh Information Axiom used the probability of successfully satisfying condition (p), whereas our adaptation is a probability of failure $(1 - p)$. The GQM behavior adapted dynamic complexity content for a particular time in t is then

$$C_t = -\ln(1 - \mathrm{p}_b) \tag{2}$$

or for multiple metrics

$$C_t = \sum_{i=1}^{l} \left(-\ln(1 - \mathrm{p}_{b_i})\right). \tag{3}$$

The metrics are intended to be generic responses to the questions, but adaptations may be applied for different fields of study. Similarly, the generic metric categories are provided with the expectation that further field/system/program-specific derivations will be made. For example, aspects of the mission and environment can be independent questions instead of being grouped. Adding questions on uses, scale, scope, physical, programmatic, acquisition, political, stakeholder, etc. results in creation of independent metrics. Furthermore, the metrics can be weighted based on perceived criticality or impact.

The complexity evaluation contained herein is not intended to be a substitute for a state-of-the-art system and interface design as reviewed in Section II-C. It is intended to complement it. For example, once various system and interface design alternatives have been proposed and modeled to minimize static

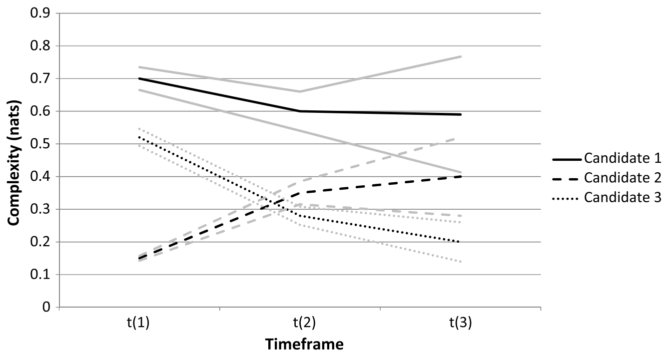

Fig. 2. Dynamic complexity versus time.

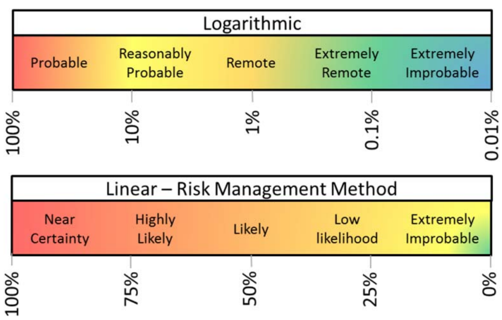

Fig. 3. Probability of occurrence definition.

complexity, the dynamic complexity evaluation may be applied to compare the final design candidates for a particular time in t

$$\begin{array}{ccc} \text{Cand 1} & \text{Cand 2} & \ldots\ \text{Cand } n \\ \sum_{i=1}^{l}\left(-\ln(1-\mathrm{p}_{b_i})\right) \Leftrightarrow & \sum_{i=1}^{l}\left(-\ln(1-\mathrm{p}_{b_i})\right) \Leftrightarrow & \ldots \end{array} \quad (4)$$

The evaluation of overall dynamic complexity may also be presented verses time in t as show in Fig. 2. The capabilities of the observer are characterized as a confidence interval around the probability point estimate. The confidence interval is plotted as error bars associated with the complexity estimate data points.

Evaluation of the probabilities can be approached using methods already used in systems engineering risk management. A log based scale may be preferable for evaluating lower likelihood conditions. Both linear and log scales are shown in Fig. 3.

C. Application Example

The proposed dynamic complexity characterization methodology can be widely applied to a broad range of system types from simple to complex. For example, the degree of difficulty in predicting future weather patterns is dependent on the specific location on the planet, the current weather patterns and how far in the future we wish to forecast. By our reference definition, weather modeling can be either simple or very complex and offers a good application example. The next few paragraphs contain a conceptual-level overview of how the proposed dynamic complexity characterization methodology would apply to weather modeling systems.

TABLE V
COMPLEXITY METRIC EVALUATION

Eval ID	P (Non-Expert)	P (Expert)	Metric
Rain-1	Ex. Rt. 0.1%	Remote 3.4%	What is the probability that precipitation levels for a location in the US West/desert cannot be predicted to within 1 inch 1 day in advance?
Rain-14	Reas. Prob. 10%	Probable 50%	What is the probability that precipitation levels for a location in the US West/desert cannot be predicted to within 1 inch 14 days in advance?
Rain-42	Probable 40%	Probable 60%	What is the probability that precipitation levels for a location in the US West/desert cannot be predicted to within 1 inch 42 days in advance?
Temp-1	Ex. Rt. 0.1%	Remote 1%	What is the probability that temperature for a location in the US West/desert cannot be predicted to within 5 degrees 1 day in advance?
Temp-14	Remote 1%	Reas. Prob. 13%	What is the probability that temperature for a location in the US West/desert cannot be predicted to within 5 degrees 14 days in advance (starting Apr 16, 2015)?
Temp-42	Reas. Prob. 10%	Reas. Prob. 6%	What is the probability that temperature for a location in the US West/desert cannot be predicted to within 5 degrees 42 days in advance (starting Apr 16, 2015)?

Precipitation and temperature for three forecast durations (1 day, 2 weeks, 6 weeks) are considered for an example region (Southwestern Arizona—desert) starting in April 2015. Entropy metrics are derived using the behavior entropy model (see Table IV). To demonstrate impact of observer capability, metrics evaluations are performed by a nonexpert first (the author's approximation) followed by an evaluation using interpolated National Weather Service Climate Prediction Center (NWS-CPC) data. The nonexpert evaluation was done blindly; that is before looking at the NWS-CPC data. The NWS-CPC confidence interval temperature (°F) and precipitation (inches of rain) ranges varied by outlook chart; thus, the data are interpolated for inclusion in the analysis.

Table V contains the expert and nonexpert complexity metric evaluation. The observer confidence interval for both evaluations is 90%. This is to say that we are 90% confident that the true value of the metric is in our confidence interval and the justification for the sigma is given in Table VI. The stochastic endpoints of sample mean $\bar{X}$ are computed assuming a normal distribution with $n = 25$ samples. The error bars in Fig. 4 show the 90% confidence interval about the stochastic endpoints.

Fig. 4 shows the results of the expert complexity evaluation versus time for the selected metrics. The error bars created using the 90% confidence interval data show overlap in the 1-day complexity when observer capabilities are considered. The confidence interval upper limits for the 1-day complexities are about the same where the lower limit of the temperature complexity is much lower than for rain modeling. For the 14 day and 42 day periods there is no overlap even considering the observer capabilities. The evaluation indicates that even considering observer capabilities, rain modeling within $1''$ is generally more complex than temperature modeling within 5 °F for the Southwestern Arizona desert in the period starting April 2015.

TABLE VI
SELECTED WEATHER SYSTEM ASPECTS OBSERVER 90% CONFIDENCE INTERVAL SUMMARY

Non-expert Evaluation	σ	$\bar{X}\pm$	Sigma Justification
Rain-1	1%	0.33%	The observers (authors) expect to be able to predict rain 1 day in advance based on non-expert observance of forecasts and results.
Rain-14	10%	3.3%	The observers have less experience in evaluating weather tools and models 7 days in advance
Rain-42	40%	13%	The observers have almost no experience in evaluating weather tools and models 30 days in advance
Temp-1	1%	0.33%	The observers (authors) expect to be able to predict temperature 1 day in advance based on non-expert observance of forecasts and results.
Temp-14	5%	1.6%	The observers have less experience in evaluating weather tools and models 7 days in advance
Temp-42	20%	6.6%	The observers have almost no experience in evaluating weather tools and models 30 days in advance
Expert Evaluation	**σ**	$\bar{X}\pm$	**Sigma Justification**
Rain-1	4.2%	1.4%	ForecastAdvisor.com summary of NWS data
Rain-14	64%	21%	NWS CPC Data
Rain-42	42%	14%	NWS CPC Data
Temp-1	12%	3.8%	ForecastAdvisor.com summary of NWS data
Temp-15	33%	11%	NWS CPC Data
Temp-42	28%	9.2%	NWS CPC Data

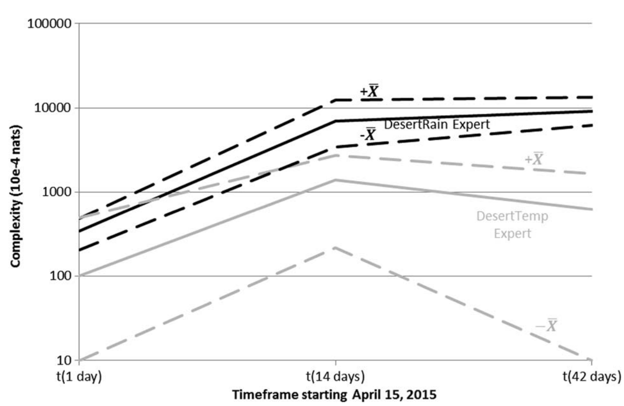

Fig. 4. Expert evaluation of selected rain and temperature weather system aspects dynamic complexity versus time.

It also indicates that the complexity of rain modeling for the selected behaviors increases more with time than do the temperature behaviors. What might seem unusual (as the nonexpert observer did not predict it), is that the 42 day temperature modeling complexity is lower than the 14 day complexity. The NWS data indicates that the longer term desert summer temperature is more predictable than the shorter term late spring temperature. If the evaluation timeframe starts in summer instead of mid-spring the outcome will be different.

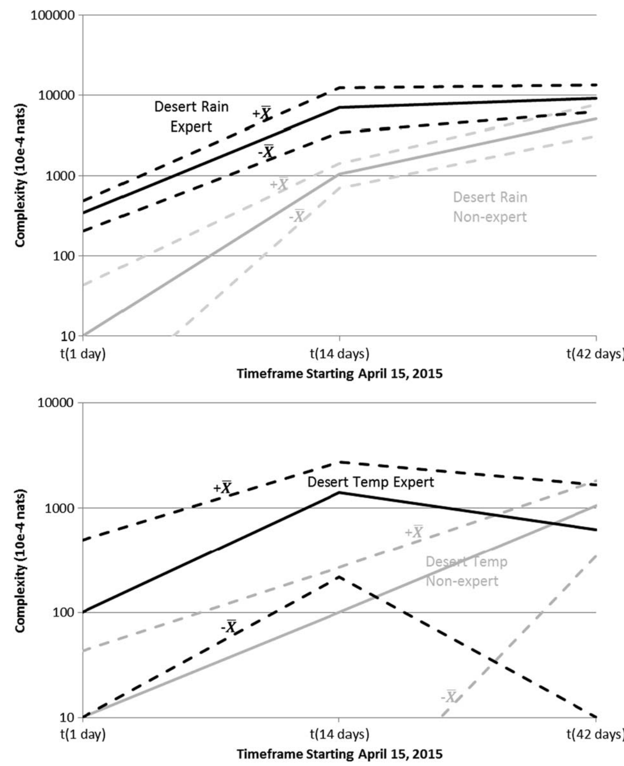

Fig. 5. Selected rain (top) and temperature (bottom) weather system aspects dynamic complexity versus time.

The results in Fig. 5 show the impact associated with varying the observer's capabilities. The expert evaluation indicates that rain and temperature complexity for the chosen region and timeframes are generally more complex than the nonexpert observer realized. The fact that the nonexpert evaluation error bars generally do not always contain the expert evaluation results indicates that the nonexpert did not even know enough to properly evaluate his own capabilities. Furthermore, the nonexpert did not know that weather complexity varies by season. The NWS-CPC charts indicate that the weather in the selected region is much more easily predicted for summer months even with a longer lead time.

This exercise demonstrates how the proposed dynamic complexity characterization methodology can readily be applied to quantitatively characterize and compare different complex systems for selected behaviors. It also shows how observer capabilities can be integrated into the complexity evaluation. Secondarily, the results emphasize the importance of expert inputs. The nonexpert evaluation demonstrates the observer's ignorance of his own ability levels.

D. Case Study

Next, the present research uses the proposed complexity metrics to evaluate a sample air traffic control radar system with two candidate architectures including only a few internal interface and subsystem differences. The example system demonstrates how the proposed complexity metrics can be applied to evaluate a system for complexity versus time in a real-world engineering program trade study. Fig. 6 shows the proposed candidate architecture introduced with the hope of

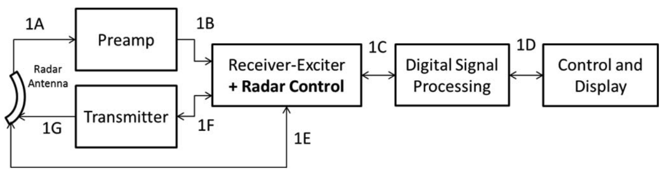

Fig. 6. Candidate system data flow.

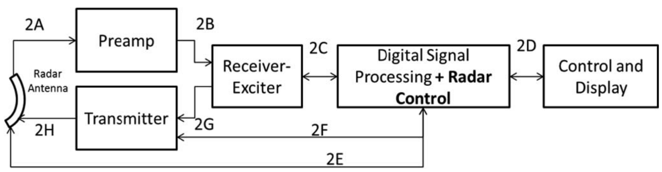

Fig. 7. Legacy system data flow.

TABLE VII
CANDIDATE SYSTEM INTERFACES AND SUBSYSTEMS

Interface ID	**Description**
1A	*n* Channels RF
1B	*n* Channels RF
1C	**10 GbE digitized data messages, GbE setup, GbE BIT/fault data**
1D	Plots, status, weather, operator control messages
1E	**Ethernet: plots, status, weather, operator control messages**
1F	Transmit status and control messages, RF exciter signal
1G	Transmit RF
Subsystem ID	**Description**
Antenna	High gain cosecant-squared dish
Preamp	*n* channel low noise amplifier
Transmitter	High power RF amplifier
Receiver Exciter; Control	***n* channel receiver, exciter, and real time hardware control**
DSP	**Signal processing, no real time control**
Ctrl/Display	User workstation

TABLE VIII
LEGACY SYSTEM INTERFACES AND SUBSYSTEMS

Interface ID	**Description**
2A	Equivalent to other candidate
2B	Equivalent to other candidate
2C	**S-FPDP digitized data, clock, discrete setup/BIT/fault data**
2D	Equivalent to other candidate
2E	**low level digital timing signals, data**
2F	Equivalent to other candidate
2G	Equivalent to subset of other candidate
2H	Equivalent to other candidate
Subsystem ID	**Description**
Antenna	Equivalent to other candidate
Preamp	Equivalent to other candidate
Transmitter	Equivalent to other candidate
Receiver Exciter; Control	***n* channel receiver, exciter, no real time control**
DSP	**Signal processing, real time hardware control**
Ctrl/Display	Equivalent to other candidate

TABLE IX
BEHAVIORAL ASPECTS SELECTED FOR APPLICATION TO CASE STUDY

What is the probability that _________ cannot be easily/accurately predicted/navigated in timeframe *t*?
Mission
$\mathbf{p_1}$: failures during manufacturing, selloff, testing, fielding; repair/replace behaviors
$\mathbf{p_2}$: aspects of the system design scale/scope
$\mathbf{p_3}$: subsystem/interface performance
Environment
$\mathbf{p_4}$: obsolescence behaviors; behavior while adding new customer features/capabilities
$\mathbf{p_5}$: development environment
$\mathbf{p_6}$: technology supportability

reducing the complexity of a legacy system (see Fig. 7) that is currently fielded.

Both the candidate and legacy systems have the same customer-level requirements and the same stakeholder environment. The candidate system uses a standard COTS Intel Xeon based server for the radar signal processing. The interface to bring the digitized radar returns into the server-based signal processor is 10-gigabit Ethernet. The legacy system uses a vendor specific set of processing cards in a VME64x standard enclosure. The interface to bring the digitized radar returns into the VME-based signal processor is Serial Front Panel Data Port (S-FPDP) standard. Tables VII and VIII highlight the similarities and differences between the candidate and legacy interfaces and subsystems.

There are only two interfaces with significant differences. For the purposes of this example, one interface (C) and one subsystem (DSP) are selected for evaluation. Interface C in the candidate system consists of 10-GbE cables, connectors, and data/messaging. Interface C in the legacy system consists of standard S-FPDP cables, connectors and data/messaging. The candidate system DSP consists of an Intel Xeon based server, Linux OS, and the ported software. The legacy system DSP consists of a vendor specific VME-based Power PC architecture, a proprietary fabric and OS, and the legacy signal software.

A metrics summary table (derived from Table IV) is given in Table IX. It provides a down-selection of the aspects of the mission and environment applied to the sample system to communicate exactly what is being evaluated. The aspects are selected to emphasize complexity differences between the candidates. In other words, the aspects that point out less predictable behaviors are down selected, whereas those that are easily predictable are ignored. In an actual development environment it can also help identify oversights in the evaluation when the entire process is reviewed upon completion.

The selected evaluation timeframes are t = (1–2 years) and t = (6–20 years). The interface complexity metrics evaluation for the selected system behavioral aspects are shown in Tables X and XI. Metrics for the selected interface and subsystem are evaluated independently for both timeframes using a logarithmic scale for probability of occurrence (see Fig. 3). The complexity metrics evaluation for the DSP subsystems are shown in Tables XII and XIII. Justification for each of the ratings is provided.

TABLE X
COMPLEXITY METRIC EVALUATION FOR INTERFACE (1–2 YEAR; C = CANDIDATE, L = LEGACY)

Metric	t=1-2 year	Justification
C p_1	Remote 1%	First manufacturing, test, selloff with new test gear and procedures. Interface has many new tests and procedures related to messaging.
L p_1	Ex. Im. 0.01%	Existing manufacturing/test/selloff procedures and gear are well understood and in production.
C p_2	Re. Pr. 10%	Updated interface hardware. Few connections and few messages, but new messaging to program real time control.
L p_2	Ex. Rt. 0.1%	Real time clocking and control across many interfaces, but current production and deployment means moderately predictable results.
C p_3	Re. Pr. 10%	The interface HW is standard and easy to predict. Messaging is an entirely new design which the observer may not be able to fully predict
L p_3	Ex. Im. 0.01%	The team has significant legacy system experience. Little public domain information available to aid predictions;
C p_4	Ex. Im. 0.01%	Obsolescence / adaptation extremely improbable in the short term.
L p_4	Ex. Im. 0.01%	Parts are available now and in the short term. Obsolescence / adaptation extremely improbable in the short term.
C p_5	Ex. Im. 0.01%	Predictable environment with common COTS interface hardware and standard software compilers and drivers
L p_5	Ex. Im. 0.01%	Aging HW, SW, and compilers, but systems with this interface are currently in use and generally sound.
C p_6	Ex. Im. 0.01%	Have experience with technology relevant to the candidate system updates, the existing system, customers, and vendors.
L p_6	Ex. Im. 0.01%	Minimal but sufficient support for interface, but limited concerns for short term predictability based on current team experience.
Ex. Im. = Extremely Improbable // Ex. Rt. = Extremely Remote // Re. Pr. = Reasonably Probable		

TABLE XI
COMPLEXITY METRIC EVALUATION FOR INTERFACE (6–20 YEAR; C = CANDIDATE, L = LEGACY)

Metric	t=6-20 year	Justification
C p_1	Ex. Im. 0.01%	Repair/replace predictable based on large market and wide use of standard .
L p_1	Re. Pr. 10%	Repair/replace parts not predictable due to possible obsolescence
C p_2	Ex. Im. 0.01%	Once messaging design stabilized, there are few connections and a robust 10 GbE design based on short term data
L p_2	Remote 1%	The high number of connections and contacts at the interface leads to unintended consequences due to vibrations, periods of storage repair actions etc
C p_3	Ex. Im. 0.01%	Widely used standard with very large market presence for better long-term predictability
L p_3	Ex. Im. 0.01%	Have experience with systems with similar interfaces which have been fielded for similar durations/environments
C p_4	Ex. Im. 0.01%	Known path for expanding interface; behaviors generally understood for host of missions/environments; operability of optics in rugged/humid environment
L p_4	Re. Pr. 10%	Difficulty in expanding interface makes sufficiency of behaviors difficult to predict.
C p_5	Ex. Im. 0.01%	Predictable environment expected for 10 GbE standard, based on history to date and expected prevalence in future
L p_5	Ex. Rt. 0.1%	S-FPDP standard and messaging environment not likely to change
C p_6	Ex. Im. 0.01%	Widely used standard: readily available support information expected in the future
L p_6	Remote 1%	Fewer internal experts with predictably small circle of support for interface
Ex. Im. = Extremely Improbable // Ex. Rt. = Extremely Remote // Re. Pr. = Reasonably Probable		

TABLE XII
COMPLEXITY METRIC EVALUATION FOR SUBSYSTEM (1–2 YEAR; C = CANDIDATE, L = LEGACY)

Metric	t=1-2 year	Justification
C p_1	Remote 1%	First manufacturing, test, selloff with new test gear and procedures. Processor has new tests and procedures related to processing SW port.
L p_1	Ex. Im. 0.01%	Existing manufacturing/test/selloff procedures and gear are well understood and in production.
C p_2	Remote 1%	new and ported software not entirely predictable, but many parts, boards, chasis replaced with few.
L p_2	Ex. Im. 0.01%	High parts count including many VME boards, chassis, power supplies. Experience indicates that results reasonably predictable in short term
C p_3	Ex. Rt. 0.1%	Have experience with the HW, technologies and software ported from the legacy system
L p_3	Ex. Im. 0.01%	Experience with legacy processor in factory/field aids predictability.
C p_4	Ex. Im. 0.01%	Obsolescence / adaptation extremely improbable in the short term.
L p_4	Ex. Im. 0.01%	Parts are available now and in the short term. Obsolescence / adaptation extremely improbable in the short term.
C p_5	Ex. Im. 0.01%	Predictable environment for COTS DSP and stable / prevalent /current operating system used.
L p_5	Ex. Im. 0.01%	This subsystem is currently in use and is operationally sound – predictable short term environment
C p_6	Remote 1%	Experience with the environment, including the OS, as applied to other programs
L p_6	Ex. Im. 0.01%	Good current experience with proprietary OS, VxWorks. Limited concerns for short term predictability based on current experience.
Ex. Im. = Extremely Improbable // Ex. Rt. = Extremely Remote // Re. Pr. = Reasonably Probable		

TABLE XIII
COMPLEXITY METRIC EVALUATION FOR SUBSYSTEM (6–20 YEAR; C = CANDIDATE, L = LEGACY)

Metric	t=6-20 year	Justification
C p_1	Ex. Im. 0.01%	Repair and replace predictable based on size/prevalence of server market
L p_1	Re. Pr. 10%	Repair and replace difficult to predict due to possible obsolescence.
C p_2	Ex. Im. 0.01%	After short term SW stabilization, simple server system is predictable in nature. Software basis (ported) fielded for many years.
L p_2	Remote 1%	The high number of connections and contacts in the VME leads to unintended consequences due to vibrations, long periods of storage repair/replace actions, etc. Cabling, seating, cycling issues difficult to predict.
C p_3	Ex. Im. 0.01%	Widely used processing architecture with very large market presence for better long-term predictability
L p_3	Remote 1%	Have experience with systems with similar processing architectures which have been fielded for similar durations/environments
C p_4	Ex. Im. 0.01%	Widely used processing architecture: known paths for adding capability to adapt missions; behaviors generally understood for host of missions/environments; operability in physical environments established
L p_4	Re. Pr. 10%	Difficulty in adding features inherent to architecture; increasing difficulty finding engineering expertise to design/modify this architecture; behavior in physical environments well understood
C p_5	Ex. Im. 0.01%	Predecessor operating systems fielded for long periods with predictable results
L p_5	Ex. Rt. 0.1%	VME, proprietary OS, and other standards not likely to change
C p_6	Ex. Im. 0.01%	Widely understood architecture and OS. Readily available support information expected in the future
L p_6	Re. Pr. 10%	Fewer internal experts with predictably small circle of support for interface. Relying on few vendors to aid predictions. Shrinking circle of support for architecture.
Ex. Im. = Extremely Improbable // Ex. Rt. = Extremely Remote // Re. Pr. = Reasonably Probable		

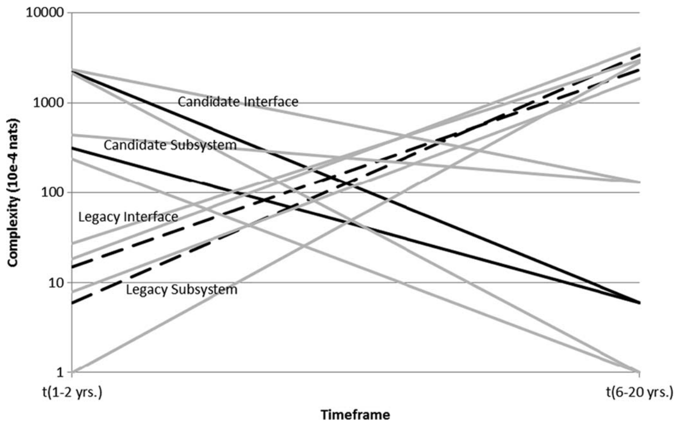

Fig. 8. Dynamic complexity versus time.

TABLE XIV
INTERFACE AND SUBSYSTEM METRICS SUMMARY FOR T = (1–2 YEARS) AND T = (6–20 YEARS)

Interface 1-2 Year	p_1	p_2	p_3	p_4	p_5	p_6	$C_{t=1\text{-}2\ yrs}$
Candidate	1%	10%	10%	0.01%	0.01%	0.01%	0.22
Legacy	0.01%	0.1%	0.01%	0.01%	0.01%	0.01%	0.0015
Interface 6-20 Year	p_1	p_2	p_3	p_4	p_5	p_6	$C_{t=6\text{-}20\ yrs}$
Candidate	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.00060
Legacy	10%	1%	0.01%	10%	0.1%	1%	0.23
Subsystem 1-2 Year	p_1	p_2	p_3	p_4	p_5	p_6	$C_{t=1\text{-}2\ yrs}$
Candidate	1%	1%	0.1%	0.01%	0.01%	1%	0.031
Legacy	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.00060
Subsystem 6-20 Year	p_1	p_2	p_3	p_4	p_5	p_6	$C_{t=6\text{-}20\ yrs}$
Candidate	0.01%	0.01%	0.01%	0.01%	0.01%	0.01%	0.00060
Legacy	10%	1%	1%	10%	0.1%	10%	0.34

The dynamic complexity computation (2) is evaluated independently for each interface and subsystem so that complexity versus time can be evaluated and presented for comparison and discussion. The complexity versus time is plotted in Fig. 8 using C(t = 1–2 years, t = 6–20 years) for the legacy and candidate interfaces and subsystems from Table XIV. The legacy interface and subsystem demonstrate a relatively low t = 1–2 year complexity which increases significantly in the t = 6–20 year timeframe. The candidate interface and subsystem demonstrate a relatively high t = 1–2 year complexity which decreases significantly in the t = 6–20 year timeframe. The observer is evaluated using a 70% confidence range of values (interval) and the justification for the sigma is given in Table XV. The stochastic endpoints of sample mean $\bar{X}$ are computed assuming a normal distribution with $n = 25$ samples. The error bars in Fig. 8 show the 70% confidence interval about the stochastic endpoints.

The case study explored demonstrates how dynamic complexity metrics can be applied to evaluate system complexity versus time in a real-world engineering program trade study. Stakeholders can use the information to aid in decision making between options. The results indicate increasing complexity versus time for the legacy interface/subsystem and decreasing complexity versus time for the candidate interface/subsystem (see Fig. 8).

TABLE XV
OBSERVER 70% CONFIDENCE INTERVAL SUMMARY

Evaluation	Time Interval	σ	$\bar{X}\pm$	Sigma Justification
Legacy	t=1-2 years	0.1%	0.02%	Observer confidence for selected behaviors backed by experience with systems currently fielded and in use. The legacy system behaviors evaluated are generally well understood. The evaluation team is current and expert. Historically team has good visibility in the 1-2 year parts/obsolescence timeframe.
Candidate	t=1-2 years	0.5%	0.10%	The candidate system involves new hardware design and test aspects, but the evaluation team is familiar with the technologies as used on related systems. The evaluation team has worked with the OS and programming environment on other similar systems. The evaluation team has less experience with the new type of messaging protocol and real-time control.
Legacy	t=6-20 years	5.0%	1.04%	The market is relatively small, so it is hard to precisely predict availability of parts. It is hard to predict availability of future expertise needed for upgrades since there may be a shrinking support circle. The observer has relatively more difficulty predicting the longer-term repair/replace actions for the legacy architecture due to the significantly higher number of parts, connections and contacts.
Candidate	t=6-20 years	1.0%	0.21%	Prediction confidence is relatively higher than with legacy evaluation due to large market presence of candidate components and wide use of standard enhancing predictability for selected aspects. High confidence that the candidate standards will endure and expand into the selected time interval. Longer term upgrade paths are well understood. Long term operability in physical environments understood from evaluation team experience with different programs

The legacy interface and subsystem (2C and DSP2) have advantages in $t = 1 - 2$ year short-term complexity since the radar system is currently in production, the observer has current experience with the components, and the short-term behaviors are expected to be predictable based on recent fielded system behaviors. In the $t = 6 - 20$ year long term, however, the small market for the candidate hardware/software/skills, potential obsolescence and general inability to expand the system leads to difficulty in predicting the future mission, behaviors, and observer capabilities.

The proposed modification candidate interface and subsystem (1C and DSP1) have disadvantages in $t = 1 - 2$ year short-term complexity since the modified interface and subsystem introduce new aspects that are not as predictable as those for the legacy example. The candidate subsystem has less predictability in its system behavior and observer capability when compared with the legacy example. The $t = 6 - 20$ year long-term story, however, is quite different. Once the initial development unpredictability is resolved, the longer term predictability is expected to be much better for the candidate interface and subsystem. Driving the relatively better predictability are components that will be widely used for many years, expertise that will be readily available in the future, and standards that have already achieved very broad acceptance. In addition, the legacy system has two subsystems with real-time clocking which must be handled across interfaces. The candidate system has only one subsystem handling real-time clocking yielding a lower system complexity-measure.

IV. Discussion of Results

How do the case study dynamic complexity-based metrics differ from structural complexity-based metrics evaluated in prior works? Structurally, the quantity of interfaces was reduced in the candidate architecture, whereas the number of interface types remained the same. The number and number of types of subsystems did not change. The quantities would have pointed to only a slightly different complexity, but the aspect of changing complexity versus time and the detailed justifications for the metrics would be missing from a structural-based complexity analysis. Dynamic complexity metrics are necessary to evaluate system complexity.

With respect to the individual or team adapting or applying dynamic complexity metrics, the application example demonstrated that expertise is critical. Unlike static complexity metrics, visibility into areas difficult to predict is challenging and should be assigned to the most experienced individuals. Selecting the wrong focus can prevent one from evaluating other things—which may be the critical things [5]. In addition, worthy of discussion is the expectation that expert customer and development team dynamic complexity metric evaluations may produce different results. Different teams (observers) may have better visibility into different aspects of the system or its behavior. This is not a problem with the method, but is indicative of the need for convergence of understanding between the teams. Use of the method proposed herein will help expose the different views for resolution. The metrics inherently focus the observers to evaluate specific key areas to reduce general bias. Having more than one expert observer converge in key areas will help ensure that the results are accurate. Divergent results should be investigated since they may represent areas requiring further trades. In this vein, it is likely beneficial for the dynamic complexity metric development team to seek outside expertise.

There are limitations when applying the proposed dynamic complexity metrics. The probability metrics in the application example and case study are evaluated with respect to other similar systems. As has already been stated, complexity calculations depend on the quantity of evaluated components, evaluator judgment, and are system, observer, and behavior dependent. Consequently, the complexity metric may be applied in a comparative evaluation within the same context. Use of the metric in an absolute sense would likely be misguided.

A further limitation of dynamic complexity measures in general is that as a system becomes more complex, we simply cannot fully understand all of the knowledge and ignorance measures. The reference definition of complexity as "the degree of difficulty in accurately predicting future behavior" indicates that there will be inherent inaccuracies in predicting the behavior of systems with higher complexity. Since the difficulty of accurately predicting future behavior increases with complexity (by our definition), the accuracy of our complexity measures decreases as the complexity of a system increases. This also makes it difficult to compare the relative complexity of highly complex systems. What this research proposes is that we can best characterize and quantify the system dynamic complexity using expertly determined knowledge and ignorance measures.

Finally, a word on behaviors. There may be core complexity associated with a system; thus, complexity will not necessarily always decrease with time/understanding. For example, in the case that different external environments can change the dynamics it is possible that the system may always have a degree of unpredictability. There may be other aspects of a system that may become less complex with time as we are able to better understand how the system behaves.

V. Conclusion

Use of a complexity framework consisting of state-of-the-art complexity-based system design can help quantitatively demonstrate overall system fragility of competing designs and modifications. This paper proposes dynamic complexity measures which have the potential to improve the relevance of complexity-based system design techniques beyond what is possible with static complexity measures alone.

Looking at the case study results from an engineering intuition viewpoint, the results align with common sense. An experienced systems engineer using the definitions presented herein and carefully examining the system should come to generally similar (albeit nonquantitative) conclusions regarding the complexity trending. While this might seem to defeat the purpose for desiring a metric-based quantitative framework, it does not. Larger scale complex systems will not be so easy to mentally decompose without the proposed framework. The weather system application example results were not as intuitive, at least from the perspective of the nonexpert observer.

Future case studies will explore systems with dependencies on different external environments and larger scale systems. We have not introduced a mechanism for normalizing based on the number of metrics which may also be investigated. Future studies may explore weighing of metrics based on perceived criticality/impact. The crossover time and the rates that complexity change (see Fig. 8) are also worthy of future investigation.

REFERENCES

[1] J. Wade and B. Heydari, "Complexity: Definition and reduction techniques," presented at the Complex Systems Design Management Conf., Paris, France, 2014, Paper 18.

[2] H. Sillitto, "On systems architects and systems architecting: Some thoughts on explaining the art and science of system architecting," in *Proc. 21st Annu. INCOSE Int. Symp.*, Singapore, Jul. 20–23, 2009, pp. 970–985.

[3] R. Dodder and R. Dare, "Complex adaptive systems and complexity theory: Inter-related knowledge domains," ESD.83: Research Seminar in Engineering Systems, MIT, Cambridge, MA, USA, 2000.

[4] P. Erdi, *Complexity Explained*. Berlin, Germany: Springer-Verlag, 2008.

[5] G. Sargut and R. McGrath, "Learning to live with complexity," *Harvard Bus. Rev.*, vol. 89, no. 9, pp. 68–76, Sep. 2011.

[6] ESD Symposium Committee, "ESD terms and definitions version 16," MIT Eng. Syst. Division, Cambridge, MA, USA, Feb. 2007.

[7] S. A. Sheard and D. A. Mostashari, *A Complexity Typology for Systems Engineering*. Hoboken, NJ, USA: Stevens Institute of Technology, 2010.

[8] R. Stevens, *Engineering Enterprise Systems: Challenges and Prospects*. McLean, VA, USA: The MITRE Corporation, 2006.

[9] J. S. Shiner and M. Davison, "Simple measure for complexity," *Phys. Rev. E, Stat. Nonlinear Soft Matter Phys.*, vol. 59, no. 2, pp. 1459–1464, Feb. 1999.

[10] T. Yamano, "A statistical complexity measure with nonextensive entropy and quasi-multiplicativity," *J. Math. Phys.*, vol. 45, no. 5, pp. 1974–1987, May 2004.

[11] D. Stuart, R. Mattikalli, D. DeLaurentis, and J. Shah, "Complexity and adaptability," DARPA META II CODA, Sep. 2011.

[12] K. Willcox, D. Allaire, J. Deyst, C. He, and G. Sondecker, *Stochastic Process Decision Methods for Complex Cyber-Physical Systems*. MIT, Cambridge, MA, USA, Jul. 2011.

[13] B. Murray *et al.*, "Complex Systems Design and Analysis (CODA) final report," United Technol. Corp., Hartford, CT, USA, Aug. 2011.

[14] J. Yoshida, M. Cowdin, T. Mize, R. Kellogg, and D. Bearden, "Complexity analysis of the cost effectiveness of PI-led NASA science missions," in *Proc. IEEE Aerosp. Conf.*, 2013, pp. 1–14.

[15] N. P. Suh, "Designing-in of quality through axiomatic design," *IEEE Trans. Rel.*, vol. 44. no. 2, pp. 256–264, Jun. 1995.

[16] R. Khire, S. Becz, H. Reeve, and L. Zeidner, "Assessing performance uncertainty in complex hybrid systems," United Technol. Res. Center, Hartford, CT, USA, 2010.

[17] W. Stevens and M. Constantine, "Structured design," *IBM Syst. J.*, vol. 13, no. 2, pp. 115–139, Jun. 1974.

[18] L. M. Laird and C. Brennan, *Software Measurement and Estimation: A Practical Approach*. Hoboken, NJ, USA: Wiley, 2006.

[19] G. C. R. Basili, H. Dieter, and V. Rombach, "The goal question metric approach," Univ. Maryland, College Park, MD, USA, 1994.

Jonathan Fischi received the B.Sc. and M.S. degrees in electrical engineering from Syracuse University, Syracuse, NY, USA, in 1994 and 1997, respectively. He is currently working toward the Ph.D. degree in systems engineering in the School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ.

Over the last 20+ years, he has performed a variety of systems engineering roles with Lockheed Martin Corporation.

Roshanak Nilchiani received the B.Sc. degree in mechanical engineering from Sharif University of Technology, Tehran, Iran, in 1998, the M.S. degree in engineering mechanics from the University of Nebraska-Lincoln, Lincoln, NE, USA, in 2001, and the Ph.D. degree in aerospace systems from the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, in 2005.

She has joined Stevens Institute of Technology as an Assistant Professor in 2006. Her research focuses on computational modeling of complexity and system "ilities" for space systems and other engineering systems, including the relationship between system complexity, uncertainty, emergence, and risk. The other track of her current research focuses on quantifying, measuring, and embedding Resilience and sustainability in large-scale critical infrastructure systems. Her research has been supported by DARPA, DoD, and DHS. Prior to Stevens, she was a Technical Consultant with 4Frontiers Corporation, a pioneer start-up space company (2005–2006). From 2002–2005, she was a Doctoral Research Assistant with Space Systems Policy and Architecture Research Consortium (SEAri Lab) at MIT focusing on decision-making and design under uncertainty, within the context of space systems. At MIT, she has also served as Mission Analyst Consultant for a mars mission project in 2003. She has been Reviewer for the IEEE systems journal and Wiley Systems Engineering Journal and has about 40 Journal and conference papers published. He is an Assistant Professor of systems engineering and has published in IEEE systems Journal. He is currently an Assistant Professor of Systems Engineering with the School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ.

Dr. Nilchiani is an Associate Member of American Institute of aeronautics and Astronautics (AIAA), member of Society of Women Engineers (SWE), New York Academy of Sciences, and INCOSE (International Council on Systems Engineering).

Jon Wade received the SB, SM, EE, and Ph.D. degrees in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA, USA, in 1984, 1986, and 1988, respectively.

He is a Distinguished Research Professor with the School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA, and currently serves as the Chief Technology Officer for the Systems Engineering Research Center (SERC), where he is leading research in the use of technology in systems engineering education and complex systems. Previously, he was the Executive Vice President of Engineering with International Game Technology, where he managed corporate wide research and development. He spent 10 years at Sun Microsystems during which time he managed the development of the UltraSPARC V based Enterprise Server family. Prior to this, he led advanced development of supercomputer systems at Thinking Machines Corporation.