

SYM-AM-16-062



PROCEEDINGS OF THE THIRTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

THURSDAY SESSIONS VOLUME II

A Complex Systems Perspective of Risk Mitigation and Modeling in Development and Acquisition Programs

Roshanak Rose Nilchiani, Associate Professor, Stevens Institute of Technology
Antonio Pugliese, PhD Student, Stevens Institute of Technology

Published April 30, 2016

Approved for public release; distribution is unlimited.

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



ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

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

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



ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Panel 16. Improving Governance of Complex Systems Acquisition

Thursday, May 5, 2016	
11:15 a.m. – 12:45 p.m.	<p>Chair: Rear Admiral David Gale, USN, Program Executive Officer, SHIPS</p> <p><i>Complex System Governance for Acquisition</i> Joseph Bradley, President, Leading Change, LLC Polinpapilinho Katina, Postdoctoral Researcher, Old Dominion University Charles Keating, Professor, Old Dominion University</p> <p><i>Acquisition Program Teamwork and Performance Seen Anew: Exposing the Interplay of Architecture and Behaviors in Complex Defense Programs</i> Eric Rebentisch, Research Associate, MIT Bryan Moser, Lecturer, MIT John Dickmann, Vice President, Sonalysts Inc.</p> <p><i>A Complex Systems Perspective of Risk Mitigation and Modeling in Development and Acquisition Programs</i> Roshanak Rose Nilchiani, Associate Professor, Stevens Institute of Technology Antonio Pugliese, PhD Student, Stevens Institute of Technology</p>



A Complex Systems Perspective of Risk Mitigation and Modeling in Development and Acquisition Programs

Roshanak R. Nilchiani—is an Associate Professor of Systems Engineering at the Stevens Institute of Technology. She received her BSc in mechanical engineering from Sharif University of Technology, MS in engineering mechanics from University of Nebraska–Lincoln, and PhD in aerospace systems from Massachusetts Institute of Technology. Her research focuses on computational modeling of complexity and systemilities 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. [rnilchia@stevens.edu]

Antonio Pugliese—received his BSc and MSc in aerospace engineering from the University of Naples Federico II in Naples, Italy, in 2010 and 2012, respectively. He also received a postgraduate Master in Systems Engineering for Aerospace Applications (SEAMIAero) from the University of Naples Federico II in 2014. He is currently a PhD student in systems engineering at the Stevens Institute of Technology in Hoboken, NJ. He worked on winged atmospheric re-entry systems, focusing on flight quality and thermal protection systems. He also has experience as a systems engineer with attitude control systems and experimental payloads for sounding rockets. [apuglies@stevens.edu]

Abstract

The current methodologies used in risk assessment are heavily subjective and inaccurate in various life cycle phases of complex engineered systems. The increase in complexity has caused a paradigm shift from root cause analysis to the search of a set of concurrent causes for each event and the relevant complexity content of the system. Many of the system's life cycle risks are currently assessed subjectively by imprecise methodologies such as color-coded risk matrix, and subsequently they suffer from unforeseen failures as well as cost and schedule overruns. This research project proposes a novel approach to major improvement of risk assessment by creating a set of appropriate complexity measures (informed by historical case studies) as pre-indicators of emergence of risks at different stages of a systems development process, and also a framework that enables the decision-makers on assessing the actual risk level at each phase of the development based on requirements, design decisions, and alternatives. The goal of this research is to capture the complexity of the system with some innovative metrics, thus allowing for better decision-making in architecture and design selections.

Introduction

Engineered systems have become progressively more complex and interconnected to other various infrastructure systems over the past few decades, and they continue to become more complex. Examples of this can be seen in various fields of engineered systems, spanning from satellites, aircrafts, and missiles to ground transportation systems and sophisticated interconnected power and communication grids. In one perspective, more complexity provides more sophisticated multi-functionality to the engineered system at hand, while in a competing perspective, concurrently can make the system more vulnerable and fragile and prone to failures and emergent behavior. The relationship between excessive complexity in design and operation of complex engineered systems to the risk, emergence, and increased manifestation of failures has been acknowledged by many experts and academics in various engineering design communities. However, there is a lack of comprehensive research that enables the discovery of the relationship between the level of complexity of a design to increased risks and failure of that system. This research is an initial study in understanding, modeling, and suggesting relevant complexity measures in



engineering design that can be used and linked quantitatively to the risk assessment of an engineered system.

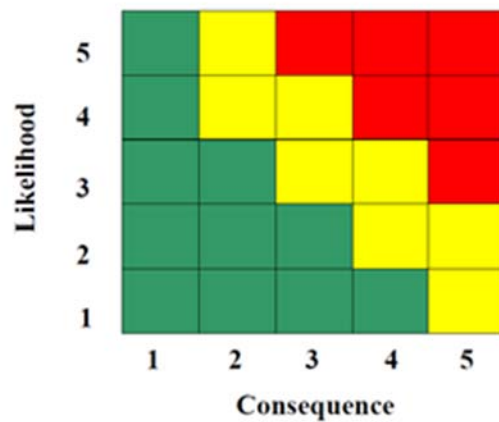


Figure 1. Traditional Risk Reporting Matrix
(DoD, 2006)

Risk can be defined as “a measure of future uncertainties in achieving program performance goals and objectives within defined cost, schedule and performance constraints” (DoD, 2006). In complex engineered systems as well as acquisition programs, often various types of risks exist that manifest themselves at different times throughout the development process. These risks can be technical, programmatic, or strategic in nature and can result in substantial cost overruns, delays, performance issues, reduced adaptability to changing requirements, or even total cancellation of a project. The major challenges of assessing risk using the traditional risk reporting matrices (Figure 1) for complex systems acquisition is that neither the likelihood nor the true consequence of a risk can be objectively established. Substantial uncertainty around the interactions among different components of a system as well as uncertainties across a multiplicity of interfaces. Also, often the symptoms and events after a failure or a problem manifest itself can be seen and are visible (Figure 2); however, the behavior and structure of the engineered system and the architecture and level of complexity of the engineered system that gives rise to such unforeseen events are often unknown. By making the complexity content and the architectural pattern of an engineered system known and explicit, in the next step of research we will be able to find the relationship between the underlying structure and complexity and the manifestation of risks and uncertainties in engineered systems.

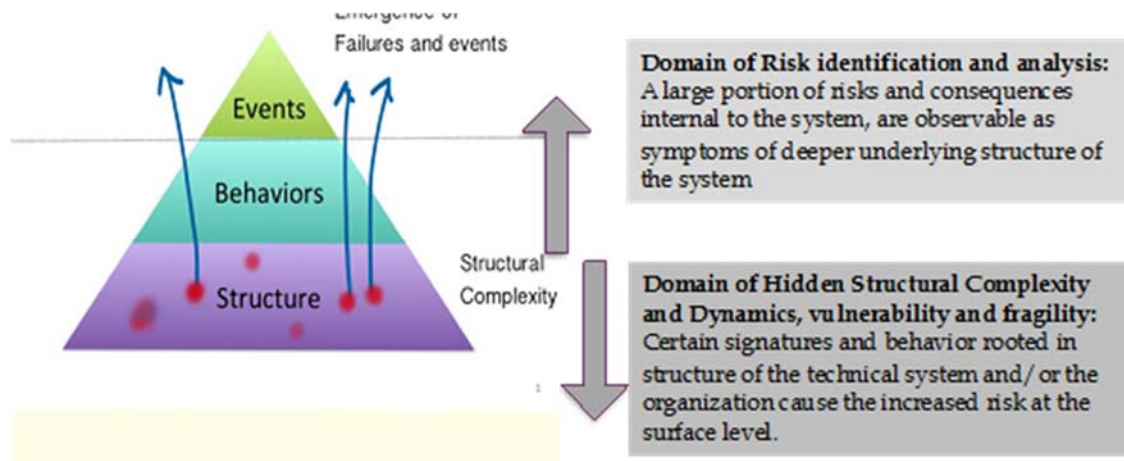


Figure 2. Problem Statement and Assessment of Structural Complexity as an Indicator of Risk and Failure Emergence

The objective of this research project is to create a quantitative and more objective assessment of technical risks and failures in engineered systems. This research aims to explore, formulate, and model the complex risks and failure mechanisms to improve the current inaccurate subjective assessment of risk in different stages of an engineered system development program as well as acquisition programs.

Literature Review

In this section of the paper, an overview of the current literature and state of the art of the complexity and complexity measurement of engineered systems as well as an overview of the literature on risk assessment of the complex engineered systems will be discussed briefly to provide a background of the current ongoing research by the authors. The literature review section begins with an overview of complex systems concepts, followed by various definitions of complexity and emergence, several current existing measures that are often being used in engineering systems designs. The section also presents a brief overview of risk assessment of complex engineered systems.

Risk Management of Complex Engineered Systems

It is not possible to know exactly how a particular design will perform until it is built. But the product cannot be built until the design is selected. Thus, design is always a matter of decision making under conditions of uncertainty and risk. (Hazelrigg, 1998)

Risk and uncertainty are the hallmarks of all complex engineered systems. The Department of Defense in the *DoD Risk Management Guide* (DoD, 2006) defines risk as follows:

Risk is a measure of future uncertainties in achieving program performance goals and objectives within defined cost, schedule and performance constraints. Risk can be associated with all aspects of a program (e.g., threat, technology maturity, supplier capability, design maturation, performance against plan). ... Risk addresses the potential variation in the planned approach and its expected outcome.

In general, risks have three components, which are the root cause, a probability (or likelihood) assessed at the present time of the root cause occurring, and the consequence (or effect) of occurrence. Often a root cause is the most basic reason for the presence of a risk. Accordingly, risks should be tied to future root causes and their effects (DoD, 2006).

In any complex technical engineering project, risk can be classified as either of technical or programmatic nature, the former concerning performance criteria and the latter focusing on cost and schedule. Both types of risk are often modeled as the product of the probability of an event and its severity (Pennock & Haimes, 2002). In modeling risk, one can also consider the future root cause (yet to happen) of a certain event (Nilchiani et al., 2013), which is where one is supposed to act in order to eliminate a specific risk. Severity and probability are traditionally represented on the widely utilized, color-coded, risk matrix. Figure 1 shows a color-coded risk matrix. Unfortunately, this seemingly quantitative tool hides subjectivity in the estimation of event frequency and severity, and for those reasons is “inapt for today’s complex systems” (Hessami, 1999). This not only means that most of the systems that we build today cannot be built with the tools and processes from last century, but also that we have started building in a domain where structural patterns matter, especially for large projects.

Complex Systems

Complexity has been one of the characteristics of many large-scale engineered systems of the past century. Complex engineered systems can provide sophisticated functionality as one side of the coin, and the other side can cause the system to be more prone to unwanted emergent behaviors and more fragility to the engineered system. The field of complexity is rich and spans over the past half century in various fields of knowledge ranging from biological systems to cyber-physical systems. As it has been discussed by several researchers, a strong correlation can be observed between the complexity of the system and various ranges of failures, including catastrophic failures (Cook, 1998; Bar-Yam, 2003; Merry & Kassavin, 1995).

In 1948, Warren Weaver, a pioneer in classifying and defining complexity in systems, described three distinct types of problems: problems of simplicity, problems of disorganized complexity, and problems of organized complexity (Weaver, 1948).

According to Weaver (1948), problems of simplicity are the problems with a low number of variables that have been tackled in the 19th century. An example is the classical Newtonian mechanics, where the motion of a body can be described with differential equations in three dimensions. In these problems, the behavior of the system is predicted by integrating equations that describe the behavior of its components. In the same article, Weaver discusses that problems of disorganized complexity are the ones with a very large number of variables that have been tackled in the twentieth century. The most immediate example is the motion of gas particles, or as an analogy the motion of a million balls rolling on a billiard table. The statistical methods developed are applicable when particles behave in an unorganized way and their interaction is limited to the time they touch each other, which is very short. In these problems it has been possible to describe the behavior of the system without looking at its components or the interaction among them.

Problems of organized complexity are the ones that are to be tackled in the 21st century, and ones that see many variables showing the feature of organization. These problems have variables that are closely interrelated and influence each other dynamically. This high level of interaction that gives rise to organization is the reason that these problems cannot be solved easily. Weaver described them as solvable with the help of powerful



calculators, but today's technology is not yet able to solve the most complex of these problems. These are the problems that nowadays we define as "complex."

Predicting the behavior of a system with many interconnected parts changing their behavior according to the state of other components is a problem of organized complexity, and the system itself is referred to as a complex system.

Cotsaftis (2009) gives a way of determining whether a system is simple, complicated, or complex by looking at its network model (i.e., nodes and edges). The model defines three types of edges: a free flight state vertex V_{ii} , a driven state from outer source vertex V_{ie} , and an interactive state with other system components vertex V_{ij} . The edges are channels along which there is a resource flux p_{ii} , p_{ie} , or p_{ij} . When

$$p_{ii} \gg p_{ij}, \inf p_{ij}, \quad (1)$$

the i^{th} component is weakly coupled with the others, external and internal. The dynamics of the component can in this case be considered independent from the other components. If the majority of the components satisfy inequality (1) the system is considered to be simple. When

$$p_{ie} \gg p_{ii}, \inf p_{ij}, \quad (2)$$

the i^{th} component is depending on outside sources. The system can still be partitioned in a set of weakly connected subsystems which dynamics is determined from outside sources. If the majority of the components satisfy inequality (2) the system is considered to be complicated. When

$$\inf p_{ij} \gg p_{ii}, p_{ie} \quad (3)$$

the i^{th} component is strongly connected to the others, and its dynamics cannot be determined without considering the effects of the other components. Also, the manipulation of the system cannot be performed as in the previous cases, since the internal connections create conditions that reduce the number of degrees of freedom. A system with a reduced number of external control dimensions that satisfies inequality (3) is said to be complex.

This definition is rather qualitative, since not all the nodes in the system have the same importance (in terms of connection number and intensity) and therefore it makes no sense to consider the majority. For this reason Cotsaftis defines the index of complexity as $C_S = n/N$, where n is the number of components that satisfy inequality (3) and N is the total number of components. A complicated system has $C_S = 0$. $C_S = 1$ corresponds to the most complex system possible, but it is also a system where external connections are negligible, and therefore the system is isolated. This is due to the fact that a complex system is describable with a low number of parameters if seen from outside, but has high connectivity in its internal structure.

Considering as an example a sheepdog and a herd of cattle, we realize that the dog has only two degrees of freedom while the herd has $2n$, where n is the number of animals in the herd. By pushing the cattle together, the dog increases their interactions and decreases the number of degrees of freedom of the herd to only two, therefore being able to control it.

The research from these two authors has shown us how complexity and simplicity are interrelated concepts, somehow opposite, but that can also be found in the same system at the same time, depending on the point of view. Madni made a distinction between systemic elegance, which "thrives on simplicity through minimalistic thinking and parsimony" and perceived elegance, which "hides systemic or organizational complexity from the user." If the system is considered to be complex but its complexity can be somehow hidden or



resolved, thus making it simpler, then the design can be considered elegant (Madni, 2012). Therefore, in order to achieve a more elegant design, we need to decrease the complexity of the system.

Emergence

Emergence is a major phenomenon related to complex engineered systems. Emergence at the macro-level is not hard-coded at the micro-level (Page, 1999). One example of emergence in natural systems is wetness. Water molecules can be arranged in three different phases (i.e., solid, liquid, and gas), but only one of them expresses a particular type of behavior, which is high adherence to surfaces. This behavior is due to the intermolecular hydrogen bonds that affect the surface tension of water drops. These bonds are also active in the solid and liquid phase, but in those cases they are either too strong or too weak to generate wetness. In this case, the emergence of a property, such as wetness, has been explained at a lower level by looking at the molecules that make up the liquid.

According to Kauffmann (2007), two different types of emergence exists (Kauffman, 2007). The reductionist approach sees emergence as epistemological, meaning that the knowledge about the systems is not yet adequate to describe the emergent phenomenon, but it can improve and explain it in the future. This is the case of wetness, where knowledge about molecules and intermolecular interactions has explained the phenomenon. On the other hand, there is the ontological emergence approach, which says that “not only do we not know if that will happen, [but] we don’t even know what can happen,” meaning that there is a gap to fill not only about the outcome of an experiment (or process), but also about the possible outcomes.

Longo presents this view with the example of the swimming bladder in fishes (Longo, Montevil, & Kauffman, 2012). An organ that gives neutral buoyancy in the water column as its main function, also enables the evolution of some kinds of worms and bacteria that will live in it. Ontological (or radical) emergence is given by the enormous amount of states the system could evolve into. In these cases we not only are not able to predict which state will happen, but we do not even know what the possible states are.

Gell-Mann also pointed out this difference using the concept of logical depth (Gell-Mann, 1995). When some apparently complex behavior can be expressed with simpler laws that reside at a lower level (e.g., the complicated pattern of energy levels of atomic nuclei that can be described at the subatomic level), the phenomenon is said to have a substantial amount of logical depth.

In our research, the emergence that is going to be tackled is considered to be epistemological emergence, logical depth according to Gell-Mann, where knowledge about the system organizational patterns and internal structure can lead to the explanation of certain phenomena. Unfortunately this concept is not so common in the systems engineering and risk management fields, and therefore this research adopts the industry jargon by talking about complexity and complex systems, but always reminding that we are actually trying to unravel logical depth from a systems engineering perspective.

Definitions and Measures of Complexity

There are various definition of complexity that have roots in various fields spanning from mathematics and biology to engineering design. In a recent paper, Wade (2014) suggests that existing complexity definitions belong to one of three types: behavioral, structural, or constructive. Behavioral definitions view the system as a black box and the measures of complexity are given based on the outputs of the system. Structural definitions look at the internal structure or architecture of the system. Constructive definitions see



complexity as the difficulty in determining the system outputs (Wade & Heydari, 2014). In this research we are interested in the modeling behavioral and structural complexity metrics. A summary of behavioral complexity definition as well as structural complexity and some measures are presented in the following sections of the literature review.

Behavioral Complexity Definitions and Metrics

The most famous behavioral complexity metric is with no doubt Shannon's entropy (Shannon, 1948). This metric evaluates the complexity by measuring the entropy of the output message of the system (this metric was initially applied to information systems).

Gell-Mann used Shannon's entropy to define information measure as a metric capable of measuring both the effective complexity, which is the amount of information necessary to describe the identified regularities of an entity, and the total information, which also takes into account the apparently random features (Gell-Mann & Lloyd, 1996). Algorithmic information content and Shannon entropy are used to build this metric. The former is responsible for measuring the effective complexity (knowledge), and the latter the random parts (ignorance). This dual approach is an interesting contribution to the measurement of complexity, since it allows one to group similar entities according to their effective complexity and to measure the diversity of the ensemble as entropy.

Chaisson (2004) proposed a specific energy-based measure of complexity—more precisely, energy rate density, which is “the amount of energy available for work while passing through a system per unit time and per unit mass” (Chaisson, 2015). This metric looks at the system as a black box and measures the net energy amount entering the system. It has been evaluated for multiple entities such as galaxies, stars, planets, plants, animals, societies, and technological systems, and also has been mapped throughout their lifetime showing an increase in complexity (Chaisson, 2014).

Willcox et al. (2011) defined complexity as “the potential of a system to exhibit unexpected behavior in the quantities of interest, regardless of whether or not that behavior is detrimental to achieving system requirements.” She proposed an entropy and probability based metric:

$$C(Q) = \exp(h(X)) \quad (4)$$

where X is the joint distribution of the quantities of interest, and $h(X)$ is the differential entropy of X defined as

$$h(X) = - \int_{\Omega_X} f_x(x) \log f_x(x) dx \quad (5)$$

where Ω_X is the support of X .

Structural Complexity Definitions and Metrics

There are a few structural complexity measures in current complex engineering systems in recent decades. The metric presented by Cotsaftis (2009) is an example of structural complexity metric, since it looks at the internal structure of the system (i.e., components and interfaces).

Another structural complexity metric was presented by McCabe for software systems (McCabe, 1976). The representation of computer programs using graphs allows one to define the cyclomatic number $v(G)$ as

$$v(G) = e - n + 2p \quad (6)$$



where G is the graph, e is the number of edges, n is the number of nodes, and p is the number of connected components. This same metric has been extended to measure architectural design complexity of a system (McCabe & Butler, 1989).

Sinha presented a structural complexity metric that uses the design structure matrix (DSM) of a system to evaluate its complexity (Sinha & de Weck, 2012). The metric is evaluated using

$$C(n, m, A) = \sum_{i=1}^n \alpha_i + \left(\sum_{i=1}^n \sum_{j=1}^n \beta_{ij} A_{ij} \right) \gamma E(A) \quad (7)$$

where n is the number of components in the system, m the number of interfaces, A the DSM, α_i the complexity of each component, $\beta_{ij} = f_{ij} \alpha_i \alpha_j$ the complexity of each interface, $\gamma = 1/n$ a normalization factor, and $E(A)$ the matrix energy of the DSM. Although the proposed metric is very sophisticated, its application sees the evaluation of α_i through expert judgment, and $f_{ij} = 1$ for lack of more information (Sinha & de Weck, 2013). One interesting feature of this metric is the topological complexity $E(A)$, which represents the level of robustness and reliability of the graph network and can be easily evaluated from the DSM through singular value decomposition.

Hybrid Structural-Behavioral Complexity Framework

The goal of this research is to develop a framework for the identification of complexity level of the engineered system and architectural patterns affecting the behavior of the system and various levels of risks. The framework will be applied at the initial design phase, when system requirements are defined, and the system architecture is in its initial development (some hierarchical levels are defined but not all of them).

Our suggested framework is based on two main ideas. The first one is decomposition. According to McCabe, the complexity of a collection of unconnected control graphs is equal to the summation of their complexities (McCabe, 1976). Wade pointed out that in complex systems, reduction by decomposition cannot work since the behavior of each component depends on the behaviors of the others (Wade & Heydari, 2014). This is true for complex engineered systems, but in this research we are tackling logical depth, and therefore we assume that the reductionist approach, as described by Kauffman (2007) can be applied to the problem.

The second idea is that it is possible to measure the complexity of an entity at its boundary. We have seen that various behavioral complexity metrics have been proposed. These metrics consider the system as a black box and only take into account its output. In this research we are going to consider not the output, but the relationship between output and input, as we believe it better describes what the system does.

Framework Application Approach

In order to measure the system complexity, the framework will combine the complexity of components that make up the subsystems at various architectural levels. This combination can be performed applying a structural complexity metric, which considers the system architecture (usually represented as a DSM or adjacency matrix) and the complexity of each component at a certain hierarchical level. The complexity of a subsystem can be evaluated with this approach, assuming that the complexity of its components and its internal structure are known. The process can be repeated upwards in the hierarchy to evaluate the complexity of the system.



At this point, this framework can use all the other structural complexity metrics already available in literature. The existing complexity measures in literature assume that the complexity of each component is already known, or if that's not the case, that it can be evaluated using expert judgment or historical data. In the creation of this framework we have attempted to remove the majority of the sources of subjectivity.

Given that the architecture is not completely defined, there will be some components that are not more than black boxes. The complexity of these components can be measured with behavioral metrics. Of course, historical data about input and output of these components in past projects will be necessary in order to evaluate the metrics, but the subjectivity coming from expert judgment will be removed. Also, there is a difference between using historical data such as input and output, which for engineered systems are physical quantities, and historical data such as rate of failure, or schedule delays due to integration, which depend on the history of the systems they are derived from.

The application of this framework can be divided into five main phases:

1. The architecture needs to be defined. It is important that there is no connection between components (or functions) at different levels, or even between components that are children of different subsystems. The only type of connection allowed for the decomposition principle to be valid is between components within the same subsystem.
2. Once the architecture is defined, it is necessary to characterize the boundary of each component. The interfaces with other components within the same subsystem need to be quantitatively classified, in order to be used in a behavioral evaluation.
3. Once the interfaces are defined and characterized according to their behavior, the complexity of each black-box component can be evaluated using a behavioral complexity metric.
4. The complexity of each subsystem is then evaluated using a structural complexity metric, from the complexity of its components and information about its internal structure.
5. Once the complexity of the lowest level components (i.e., the leaves of the hierarchy tree) is evaluated, it can be combined in a bottom-up approach to evaluate the complexity of the higher level subsystems by repeating the previous steps until the complexity of the overall system is evaluated.

This framework has been built with flexibility in mind, meaning that the interface characterization model, the behavioral metric, and the structural metric are supposed to be plugged in according to the specific characteristics of the enterprise building the system, and the type of system. We have attempted to remove the majority of the subjectivity from the evaluation, since the level of accuracy depends heavily on the level of experience of the experts, but we want to retain the knowledge that any system architect has about the system that its enterprise is comfortable building. Two senior system architects are going to evaluate architectures differently, according to their experience and the experience of the people they worked with, thus naturally picking the best choice for the enterprise they work for. Just as likely, the framework can be adapted to rate as "better designed" the architectures having traits that the enterprise successfully implemented in past projects. Figure 3 shows a summary of the hybrid structural-behavioral framework.



Hybrid Structural-
behavioral complexity
assessment framework

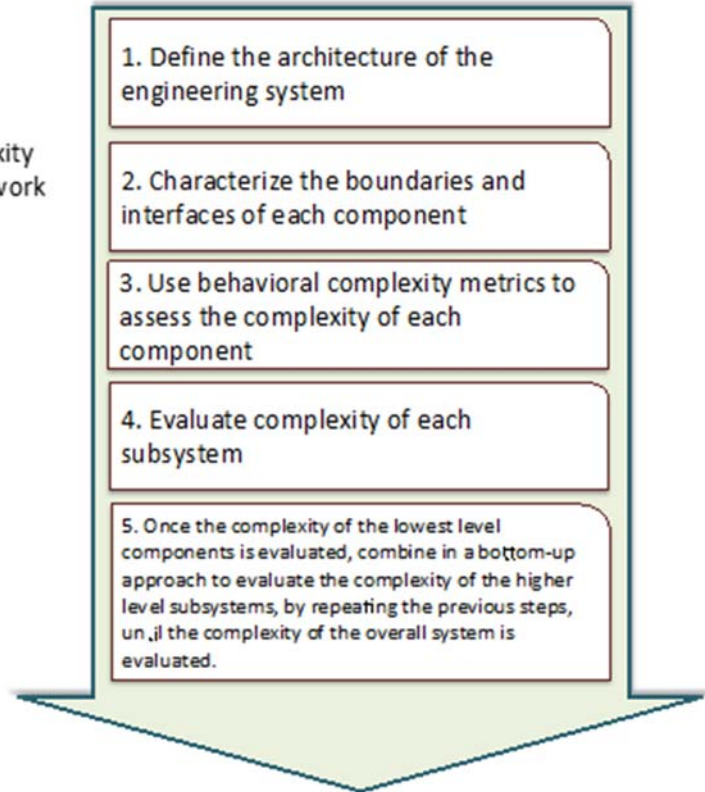


Figure 3. Schematics of the Hybrid Structural-Behavioral Complexity Assessment Framework

Part of this research effort is devoted to generating the modules (interface models, behavioral metrics, and structural metrics) that will then be used in the framework, and also to understanding which set of modules will give the best fit for each specific enterprise.

Interface Characterization Model

The connections between the components of an engineered systems are of various natures and often incommensurable. For example, considering two components having a mechanical and a thermal interface: Is it better to have low mechanical stresses and high thermal fluxes, or vice versa? In order to answer this question, the interfaces need to be classified in a scale that allows comparison between them even when they are of different natures. This will enable the evaluation of many structural and behavioral metrics that include interface complexity.

Currently this model is still under refinement. The assumptions are based on the idea that connections can be ranked in terms of how enabling they are towards a specific goal. As an example, consider the two groups of animals depicted in Figure 4.



Figure 4. Herd of Sheep and Army of Ants

Note. These two groups of animals are examples of constraining and enabling interactions.

Both the herd of sheep and the army of ants are a group of animals that interact with each other. Here the interaction of interest is the purely mechanical one. This type of interaction is constraining in the case of the herd, since it decreases the degrees of freedom of the system. This also happens in the case of the army of ants, but in this case the system has gained in capabilities (i.e., the ability to bridge in mid-air). The emergence of this capability is given by the enabling nature of the mechanical connection. The goal of this part of the research regarding interface modeling is to develop a metric for the evaluation of the level of enablement of any interface towards a specific component, within engineered systems.

Use Case: Satellite Attitude Control System

In order to show how the framework can measure the complexity of a system, we have applied the initial framework to the architecture of an Attitude Control System (ACS) for a satellite. The preliminary architecture is represented in Figure 5.

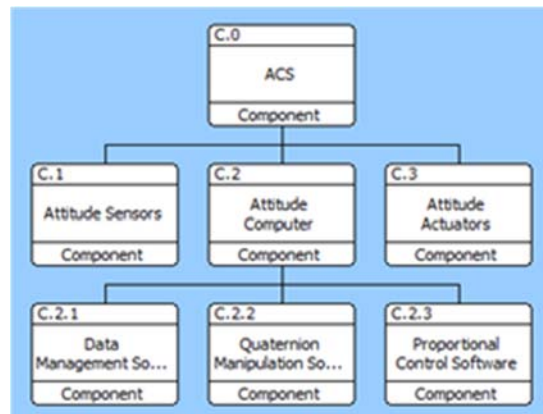


Figure 5. Hierarchical Representation of the Architecture of the ACS

The component C.0, in this case the ACS, is made up of three components—C.1, C.2, and C.3—which are the attitude sensors, attitude computer, and attitude actuators, respectively. For the sake of this example, the architecture of the component C.2 has been laid out only for its software. This architectural level includes components C.2.1, C.2.2, and C.2.3, namely data management software, quaternion manipulation software, and proportional control software. The physical architecture presented in Figure 6 has a one-to-

one mapping with the functional architecture, and therefore, for the purposes of this example, they are considered as equivalent.

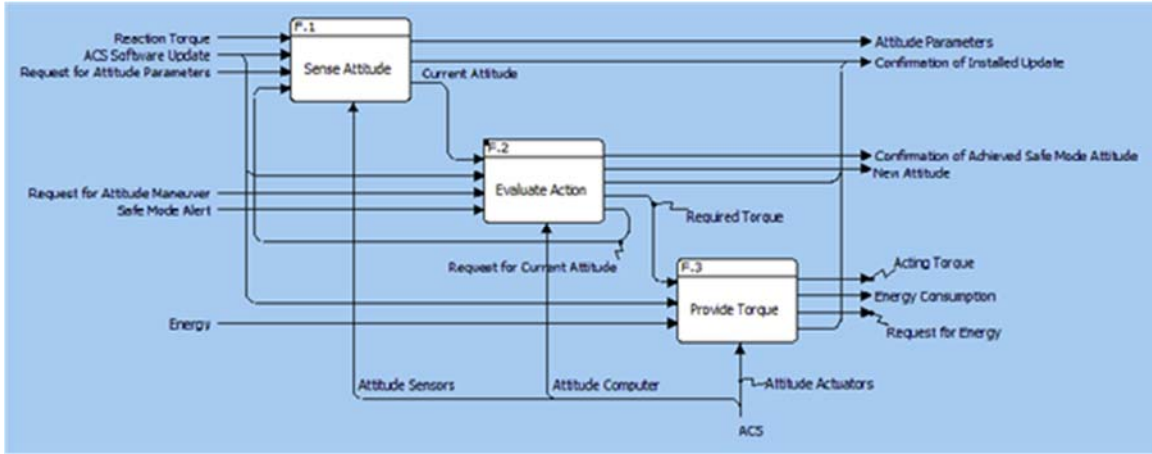


Figure 6. IDEF0 Representation of the F.0 Function Corresponding to the C.0 Component, the ACS

A hierarchical representation of the system architecture is not enough for the application of the framework. The interfaces between the components also need to be defined. Figure 6 shows these interfaces within the F.0 function. The interactions have been defined on the basis of four use cases: attitude maneuver, safe mode attitude maneuver, provide attitude parameters, and ACS software update. The information reported in Figure 6 allows us to build an adjacency matrix for the components of C.0 that can be used in the evaluation of any structural complexity metric.

$$A_{C.0} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (8)$$

In this example, the complexity metric proposed by Sinha & de Weck (2013),

$$C(n, m, A) = \sum_{i=1}^n \alpha_i + \left(\sum_{i=1}^n \sum_{j=1}^n \beta_{ij} A_{ij} \right) \gamma E(A) \quad (9)$$

will be used to evaluate the complexity of the C.0 component $C_{C.0}$. In this case $\alpha_i = C_{C.i}$, $\gamma = 1/3$ can be evaluated using singular value decomposition and taking the sum of the diagonal values $E(A_{C.0}) = 1 + \sqrt{2}$. Equation 9 then becomes

$$C_{C.0} = C_{C.1} + C_{C.2} + C_{C.3} + \frac{1 + \sqrt{2}}{3} (\beta_{12} + \beta_{21} + \beta_{23}). \quad (10)$$

Equation 10 still has many unknown variables, which need to be computed. β_{12} , β_{21} , and β_{23} can be evaluated using the interface characterization model. The evaluation of $C_{C.2}$ has the same structural approach of $C_{C.0}$, since its internal architecture has been already defined. The hybrid nature of this framework allows consideration of the most information available, evaluating the complexity of components with already defined internal structure using structural complexity metrics that take the aforementioned structured into account.

$C_{C.1}$ and $C_{C.3}$ can be evaluated using a behavioral complexity metric. This approach is necessary since these components are only defined as black boxes and we only have

information about their input and output. Evaluating the complexity of C.1 using an approach based on Chaisson's metric is taken at this stage. The metric considers the energy that the component exchanges. In the case of engineered systems, energy can be exchanged in a variety of ways (e.g., chemical, data, mechanical, thermal). The evaluation of this exchange is also part of the interface characterization model under development in this research.

In order to understand the dependency of the structural complexity on the interfaces, we can modify the architecture of F.0 by adding a connection between F.1 and F.3. In this case, the new component C.O will have an adjacency matrix:

$$A_{C.O'} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (11)$$

This leads to a different value of the matrix energy $E(A_{C.O'}) = 1 + \sqrt{3}$ and thus to a new formulation for the complexity of the component:

$$C_{C.O} = C_{C.1} + C_{C.2} + C_{C.3} + \frac{1 + \sqrt{3}}{3} (\beta_{12} + \beta_{13} + \beta_{21} + \beta_{23}) \quad (12)$$

This change in the architecture increases the complexity of the component. Other structural complexity metrics such as the metric proposed by Sinha cannot capture this change properly, since an addition of a single connection between two components leads in this case to two changes in the complexity evaluation. For this reason, in this research we will continue to propose modifications to existing complexity metrics so that the overall framework can lead to more meaningful evaluations.

Summary and Future Work

In this research we propose a framework to perform a quantitative and more objective assessment of complexity level, as a major precursor to assessing objective technical risks and failures in engineered systems. This is part of a larger research vision and objective of a theoretical model of failure mechanisms and risks in engineered systems, which is based on the complexity content of the system. This part of our research focuses on the preliminary design phase complexity assessment and follows and builds upon the previous work by Salado and Nilchiani (2012) on the complexity assessment of requirements and its translation in risks and vulnerability assessment. The new framework suggested, once completed, will be applicable to both development and acquisition programs, as long as the system architecture is partially available.

References

- Bar-Yam, Y. (2003). When systems engineering fails: Toward complex systems engineering. *IEEE International Conference on Systems, Man and Cybernetics*, 2, 2021–2028.
- Chaisson, E. J. (2004). Complexity: An energetics agenda. *Complexity*, 9(3), 14–21.
- Chaisson, E. J. (2014). The natural science underlying big history. *The Scientific World Journal*.
- Chaisson, E. J. (2015). Energy flows in low-entropy complex systems. *Entropy*, 17(12), 8007–8018.
- Cook, R. I. (1998). How complex systems fail. *Cognitive Technologies Laboratory*. Chicago IL: University of Chigaco.
- Cotsaftis, M. (2009). What makes a system complex? An approach to self organization and emergence. In *From System Complexity to Emergent Properties* (pp. 49–99). Springer.



- DoD. (2006). *Risk management guide for DoD acquisition*. Washington, DC: Author.
- Edmonds, B. (1995). What is complexity? The philosophy of complexity per se with application to some examples in evolution. *The Evolution of Complexity*.
- Gell-Mann, M. (1995). What is complexity? Remarks on simplicity and complexity by the Nobel Prize-winning author of *The Quark and the Jaguar*. *Complexity*, 1(1), 16–19.
- Gell-Mann, M. (2002). What is complexity? In *Complexity and Industrial Clusters*, 13–24. Springer.
- Gell-Mann, M., & Lloyd, S. (1996). Information measures, effective complexity, and total information. *Complexity*, 2(1), 44–52.
- Griffin, M. D. (2010). How do we fix system engineering? *International Astronautical Congress*.
- Hazelrigg, G. A. (1998). A framework for decision-based engineering design. *Journal of Mechanical Design*, 120(4), 653–658.
- Hessami, A. (1999). Risk management: A systems paradigm. *Systems Engineering*, 2(3), 156–167.
- Kauffman, S. (2007). Beyond reductionism: Reinventing the sacred. *Zygon*, 42(4), 903–914.
- Longo, G., Montevil, M., & Kauffman, S. (2012). No entailing laws, but enablement in the evolution of the biosphere. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, 1379–1392.
- Madni, A. M. (2012). Elegant systems design: Creative fusion of simplicity and power. *Systems Engineering*, 15(3), 347–354.
- McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, (4), 308–320.
- McCabe, T. J., & Butler, C. W. (1989). Design complexity measurement and testing. *Communications of the ACM*, 32(12), 1415–1425.
- Merry, U., & Kassavin, N. (1995). *Coping with uncertainty: Insights from the new sciences of chaos, self-organization, and complexity*. Praeger/Greenwood.
- Nilchiani, R., Mostashari, A., Rifkin, S., Bryzik, W., & Witus, G. (2013). *Quantitative risk—Phase 1*. Systems Engineering Research Center—University Affiliated Research Center.
- Page, S. E. (1999). Computational models from A to Z. *Complexity*, 5(1), 35–41.
- Pennock, M., & Haimes, Y. (2002). Principles and guidelines for project risk management. *Systems Engineering*, 89–108.
- Salado, A., & Nilchiani, R. (2012). A framework to assess the impact of requirements on system complexity.
- Salado, A., & Nilchiani, R. (2013a). Elegant space systems: How do we get there? In *Proceedings of the IEEE Aerospace Conference* (pp. 1–12). IEEE.
- Salado, A., & Nilchiani, R. (2013b). Using Maslow’s hierarchy of needs to define elegance in system architecture. *Procedia Computer Science*, 16, 927–936.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379–423.
- Sinha, K., & de Weck, O. L. (2012). Structural complexity metric for engineered complex systems and its application. Gain Competitive Advantage by Managing Complexity: *Proceedings of the 14th International DSM Conference Kyoto, Japan*, (pp. 181–194).
- Sinha, K., & de Weck, O. L. (2013). A network-based structural complexity metric for engineered complex systems. *IEEE International Systems Conference (SysCon)* (pp. 426–430).



- Wade, J., & Heydari, B. (2014). Complexity: Definition and reduction techniques. *Proceedings of the Poster Workshop at the 2014 Complex Systems Design & Management International Conference* (pp. 213–226).
- Weaver, W. (1948). Science and complexity. *American Scientist*, 36(4), 536–544.
- Willcox, K., Allaire, D., Deyst, J., He, C., & Sondecker, G. (2011). *Stochastic process decision methods for complex-cyber-physical systems* (Technical report, DTIC document).





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

www.acquisitionresearch.net