A fuzzy clustering methodology to analyze interfaces and assess integration risks in large-scale systems

Josh Henry Goldschmid

A FUZZY CLUSTERING METHODOLOGY TO ANALYZE INTERFACES AND ASSESS INTEGRATION RISKS IN LARGE-SCALE SYSTEMS

by

JOSH HENRY GOLDSCHMID

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

SYSTEMS ENGINEERING

2021

Approved by:

Steven Corns, Advisor
Cihan Dagli
Benjamin Kwasa
Suzanna Long
Lesley Low
Henry Pernicka

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following three articles, formatted in the style used by the Missouri University of Science and Technology:

Paper I, Pages 7-22, has been published in the Proceedings of the Complex Adaptive Systems Conference, in June 2021.

Paper II, Pages 23-58, has been published in the Systems Engineering Journal by the International Council on Systems Engineering (INCOSE) in June 2021.

Paper III, Pages 59-83, is intended for submission to the American Society of Engineering Management journal (EMJ).

ABSTRACT

Interface analysis and integration risk assessment for a large-scale, complex system is a difficult systems engineering task, but critical to the success of engineering systems with extraordinary capabilities. When dealing with large-scale systems there is little time for data gathering and often the analysis can be overwhelmed by unknowns and sometimes important factors are not measurable because of the complexities of the interconnections within the system. This research examines the significance of interface analysis and management, identifies weaknesses in literature on risk assessment for a complex system, and exploits the benefits of soft computing approaches in the interface analysis in a complex system and in the risk assessment of system integration readiness. The research aims to address some of the interface analysis challenges in a large-scale system development lifecycle such as the ones often experienced in aircraft development. The resulting product from this research is contributed to systems engineering by providing an easy-to-use interface assessment and methodology for a trained systems engineer to break the system into communities of dense interfaces and determine the integration readiness and risks based on those communities. As a proof of concept this methodology is applied on a power seat system in a commercial aircraft with data from the Critical Design Review.

ACKNOWLEDGMENTS

Completing my Ph.D. studies is harder than I thought and more rewarding than I imagined. I would like to thank Dr. Steven Corns for overseeing my Ph.D journey, his profound patience and great attitude ensured my successful completion of the program while working full-time in Washington State. He deserves more credit that I can give him. I am grateful to Dr. Cihan Dagli who was like my secondary advisor and the one who convinced me to participate in the Ph.D. program when I attended his information session at my employment location.

I would like to thank my other committee members, Drs. Benjamin Kwasa, Suzanna Long, Henry Pernicka, and Leslie Low. Dr. Low was my former boss and I thought (and still do) she was the smartest person I've met. Thank you Karen Swope for guiding me through all the Ph.D. program paperwork that at some moments I thought was more difficult than the homework assignments that Drs Corns and Dagli gave me. Thank you to the remaining faculty and staff of the Department of Engineering Management and Systems Engineering for their support.

Thank you to my parents David and Joyce for their extraordinary sacrifices and support and for not listening to whoever told them that their son's greatest achievement would be graduating from high school. You both are amazing and same goes to my sisters Renee and Michelle for being silly farts. You both rock.

Most importantly, to my beautiful, smart, and caring wife Angie, thank you so much for your love and support through my studies and for helping me become a better person. I wanna thank you for so many things so please start counting!

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

1.1. BACKGROUND AND MOTIVATION

Systems engineering enables a variety of impressive modern systems that carry hundreds of passengers across oceans, send people to the moon and artifacts on Mars, and defend nations. Sometimes realizing these systems comes at significant cost due to complexity (GAO, 2018). Every effort is made to invent and evolve tools, methods, and processes to effectively design and build complex systems with extraordinary capabilities (Beihoff et al., 2014; INCOSE, 2007; Watson, 2019). The growth of system complexity can be exemplified in the evolution of modern aircraft. The F-16 from 1974 had 15 subsystems and approximately 10^3 interfaces while the F-35 introduced in 2015 had 130 subsystems with approximately 10^5 interfaces, a 100-fold difference in the number of interfaces in 40 years (Arena, Younossi, Brancato, Blickstein, & Grammich, 2008). A few years later after the first flight of the F-35, the 787-8 Dreamliner took to the skies for the first time. The Boeing 787 was designed to have performance not previously achieved such as having larger windows, enhanced electrical systems, engines with exceptional fuel efficiency, and many components constructed with primarily composite materials (Boric, 2018; Lu, 2010; Rusnák, 2013).

The success of a complex system depends on the interactions of its components such that system as a whole is greater than the sum of its parts. This is evident in healthcare systems that depend on interoperable people, facilities, processes, services, technology, and information to dictate the patient’s treatment journey (Ahsan, K, Hanifa, S, Kingston, 2010; Al-Sakran, 2015; Kaplan, Bo-Linn; Carayon, Pronovost, Rouse, Reid, Saunders, 2013; Miles, 2009; Muhammad, Ahsan, 2016). The day-to-day healthcare conditions demand adaptation and flexibility to maintain control over a large variety of patients, flow of information, and the complexity of their symptoms. Thus, overseeing critical interfaces and managing integration risks are key to successful development and management of these and other complex systems.

The motivation of this research was to evolve systems engineering tools methods, and processes to effectively design and build highly capable and complex systems, focusing on an area that affects risks to budget and safety: interface management. (Davies, 2020; Jackson, 2016). In large scale aircraft systems development programs with a large supply chain base, Interface Control Working Groups (ICWGs) and technical review meetings becomes necessary to analyze and cooperate the interfaces (Department of Defense Systems Management College, 2001). In healthcare systems that are considered systems of systems (SoS), a quarter of the hospitals and half of the nursing homes in the United States are independent and biotechnologies are provided by thousands of small firms which can complicate interoperability needs.

Interface and integration readiness analysis for complex systems is a difficult systems engineering task, but critical to the success of engineering systems. When dealing with large-scale systems there is little time for data gathering & interface analysis and often these tasks can be overwhelmed by unknowns and important factors that are not measurable because of the complex interconnections within the system. Consequences of inadequate interface analysis is usually manifested during the systems verification phases and often experienced in the aerospace sector.

There is an inherent bias to underestimate external threats and challenges that commonly disrupt schedule and cost plans in the development of complex systems (Jaifer, Beauregard, & Bhuiyan, 2020; Reeves, Eveleigh, Holzer, & Sarkani, 2013). The external threats in this context can mean accounting changes, rigorous time frame for product development, customer expectation changes, supplier changes, program requirement changes, economic and political issues, or other forces of nature. Similarly, in the healthcare industry, new regulations, new medical technologies, new treatment options, and even new IT tools can affect the organizational structure and processes, culture, and technologies and the interoperability among them (Herzlinger, 2006). In this research we only consider the technical interfaces typically outside of a traditional hierarchical product breakdown structure that a design team may have overlooked. An example would be the electrical connections needed for fire detection in an aircraft lavatory. These external interfaces are often neglected in aircraft design (Jackson, 2016).

This research examines the significance of interface analysis and management, identifies weaknesses in literature on risk assessment, and exploits the benefit of soft computing approaches in the analysis of interfaces in a complex system and in the risk assessment of system integration readiness. The resulting product from this research is an easy-to-use interface assessment methodology for a trained systems engineer to break the system into communities of dense interfaces and determine the integration readiness and risks based on those communities. As a proof of concept this methodology is applied on a power seat system in a commercial aircraft with data from the Critical Design Review. This research explores existing strategies to mitigate interface-induced risks such as use of Design Structure Matrices and Interface Readiness Metrics. The illustrative example examines how interface issues as small a faulty Electronic Module Assembly (EMA) in a Business Class seat airbag system exacerbates 777 system vulnerabilities such as reducing the survivability of approximately 60 Business Class passengers. The methodology compliments existing integration risk mitigation strategies through a tool that provides:

- Systems engineers a complex system aggregation of communities for interface analysis, making it possible to discover missing or immature interfaces;
- A more accurate measure for system integration readiness than SRL metric;
- A perspective of the performance of interacting components within a community
- A validation of interface maturity with performance analysis;
- A risk perspective of integration scope of communities in a large-scale complex system.

The methodology uses soft computing principles explore the network within the system, aggregate system elements into communities, and uses a "community maturity level" metric assess integration readiness that is more accurate than the System Readiness Level metric. The approach then uses a fuzzy inference system to evaluate the integration risks of each community.

The contribution to systems engineering is to provide an integration readiness and risk assessment methodology that draws attention to problem areas that engineers and management need to thoroughly evaluate.

1.2. RESEARCH OBJECTIVES, APPROACH, AND CONTRIBUTION

The primary objective of this dissertation is to provide a novel approach for trained systems engineer to analyze interfaces and risks in a complex system. At the start of the research, the following questions were developed and are fundamental to the overall contributions to the Systems Engineering discipline:

1. What is my contribution to the systems engineering discipline?
2. What problem am I trying to solve?
3. What would the future look like using my approach?
4. Future work beyond my approach?

During the research I put forth the conditions of the new approach that is applicable to a real systems problem:

- The approach should draw upon the strengths of Systems Engineering and other techniques and offer unique and novel assessment approaches
- The approach should be objective and work effectively at different levels of ambiguity
- The approach should be scalable and repeatable and address different types of interface information – functional interfaces, physical, and logical or any element that exchange information.
- The approach shall be usable to support the engineering lifecycle and address any changes due to design decisions to highlight the implications of design decisions with respect to the system's maturity, and performance measures.
- The approach should provide design insight on potential missing interfaces affecting overall system performance

The major contributions of this research can be summarized as follows:

Publication I: The paper discusses the healthcare system as a Complex Adaptive System of system (SoS) and that is fragmented with independent systems and information. A System of Systems Explorer (Version 2.1.0.1 Copyright© 2017 Missouri University of Science and Technology, Systems Engineering SMART Lab) is used to to select an optimal heathcare architecture that meet key performance attributes (KPAs) based on system characteristics and system interfaces. The purpose of the study is to understand how interfaces have implications to system performance and which system to implement.

Publication II: The clustering-based interface assessment framework discussed in this paper can be used to break a complex system with highly interactive components into communities for an exhaustive integration readiness analysis. The approach was tested on a commercial aircraft seat system with data from the Critical Design Review.

Publication III: The clustering-based interface assessment framework is combined with a methodology for quantifying integration risks using a fuzzy assessor. The goal is to enable engineering managers to review the risks of each community on a 5x5 risk matrix and make decisions on risk mitigation plans.

PAPER

I. SOS EXPLORER APPLICATION WITH FUZZY-GENETIC ALGORITHMS TO ASSESS AN ENTERPRISE ARCHITECTURE – A HEALTHCARE CASE STUDY

Josh Goldschmid[a], Vinayaka Gude[b], Steven Corns[a]

[a]Missouri University of Science and Technology, Rolla, MO, USA, 65401

[b]Louisiana State University – Shreveport, Shreveport, LA USA 71115

ABSTRACT

A healthcare system is considered as a Complex Adaptive System of system (SoS) with agents composed of strategies, people, process, and technology. Healthcare systems are fragmented with independent systems and information. The enterprise architecture (EA) aims to address these fragmentations by creating boundaries around the business strategy and key performance attributes that drive integration across multiple systems of processes, people, and technology. This paper uses a SoS Explorer to select an optimal architecture that provide the necessary capabilities to meet key performance attributes (KPAs) in a dynamic, complex healthcare business environment. The SoS Explorer produces an optimal meta-architecture where all but two systems (disease and facility processes) participate with many of the systems having at least four interfaces. The healthcare meta-architecture produced in this study is not a solution to address the challenges of the healthcare enterprise architecture but provides insight on the areas – systems, capabilities, characteristics, and interfaces – to pay attention to where agility is an important attribute and not to be severely compromised.

1. INTRODUCTION

Kevin Dooley (Dooley, 1997) defined Complex Adaptive System (CAS) as a group of semi-autonomous agents who interact in interdependent ways to produce system-wide patterns, such that those patterns then influence behavior of the agents. A healthcare system recognized to have 20 industry stakeholders (Vincent & Amalberti, 2016) is considered a Complex Adaptive System of system (SoS) with agents composed of strategies, people, process, and technology (McDaniel, Lanham, & Anderson, 2009; Rouse, 2008; Wickramasinghe, Chalasani, Boppana, & Madni, 2007). A hospital has multiple types of branches, professions and varying work conditions across clinical environments such as pharmacy, nuclear medicine, radiotherapy, and blood transfusion. All of these rely heavily on automation and information technology to communicate the thread of patient information and dictate the patient's treatment journey. The day-to-day conditions demand adaptation and flexibility to maintain control over a large variety of patients, flow of information, the complexity of their symptoms and the vulnerabilities of the healthcare system. Many successful, valuable efforts to improve safety and quality of healthcare have been inspired by other industries such as aviation and nuclear, yet the complex diversity, intimacy. and sensitivity of healthcare cannot be compared to these other industries (Macrae & Stewart, 2019).

Today, healthcare systems are fragmented. A quarter of the hospitals and half of the nursing homes in the United States are independent and biotechnologies are provided by thousands of small firms (Herzlinger, 2006). The enterprise architecture (EA) aims to address the fragmentations by creating boundaries around the business strategy and drive information integration across multiple systems of processes, people, and technology (Bredemeyer, D.; Krishnan, R.; Lafrenz, A.; Malan, n.d.; Harishankar & Daley, 2011; Malta & Sousa, 2016). Since the competitive landscape changes rapidly over time, companies are forced to change their strategic objectives affecting organizational structure, culture, and the technologies used. These changes become more frequent in healthcare rendering the need for an agile enterprise architecture (Madni & Sievers, 2014) (Olsen, 2017). This paper proposes the application of the SoS Explorer utilizing computational intelligence to generate the best possible enterprise architecture providing the necessary capabilities to meet key performance attributes (KPAs).

1.1. BUSINESS AGILITY

Companies' business systems must be flexible and adaptive to cater to changing business requirements and strategies. Business agility is a key attribute to ensure the continuation of company function and performance by managing the necessary changes to adapt to both the market and technological changes (Gaona Caceres & Rosado Gómez, 2019; Hazen, Bradley, Bell, In, & Byrd, 2017). Alberts and Hayes (Alberts & Hayes, 2006) believe the key dimensions of agility have the following six attributes:

1. Robustness: the ability to maintain effectiveness across a range of tasks, situations, and conditions.
2. Resilience: the ability to recover from or adjust to misfortune, damage, or a destabilizing perturbation in the environment.
3. Responsiveness: the ability to react to a change in the environment in a timely manner.

4. Flexibility: the ability to employ multiple ways to succeed and the capacity to move seamless between them.
5. Innovation: the ability to do new things and the ability to do old things in new ways; and
6. Adaption: the ability to change work processes and the ability to change the organization.

Several researchers consider responsiveness as a key attribute of agility (Christopher & Towill, 2000; Murray, 1996; Ramasesh, Kulkarni, & Jayakumar, 2001). The theme of agility is the capability to respond and adapt to changes to meet strategic goals. In the selection of an optimal enterprise or business architecture, this paper proposes to use the following KPAs: Cycle Performance, Robustness, Flexibility, and Scalability where cycle performance is a measure of responsiveness.

1.2. ARCHITECTURE FRAMEWORK

TOGAF ADM (The Open Group Architecture Framework Architecture Development Method) is the selected architecture framework due to its recognition as a global best practice for enterprise architecture and provides flexibility and balance between IT efficiency and changes in business strategy (Kotusev, 2018). TOGAF is used by businesses to drive business goals and requirements into business infrastructures with process & tool solutions. Using TOGAF ADM to define the components of the system architecture, the initial step is to define the scope of the problem and need which is the Preliminary Phase and Architecture Vision. The vision of the healthcare architecture is to provide a system solution that ensures high healthcare service quality, increased patient satisfaction, and reduced deaths and accidents.

Figure 1 is a TOGAF model of the healthcare system from the works of Haghighathoseini, et. al. (Haghighathoseini, Bobarshad, Saghafi, Rezaei, & Bagherzadeh, 2018), where each layer – business, application, data, and technology – is linked with informational, behavioral, and structural aspects.

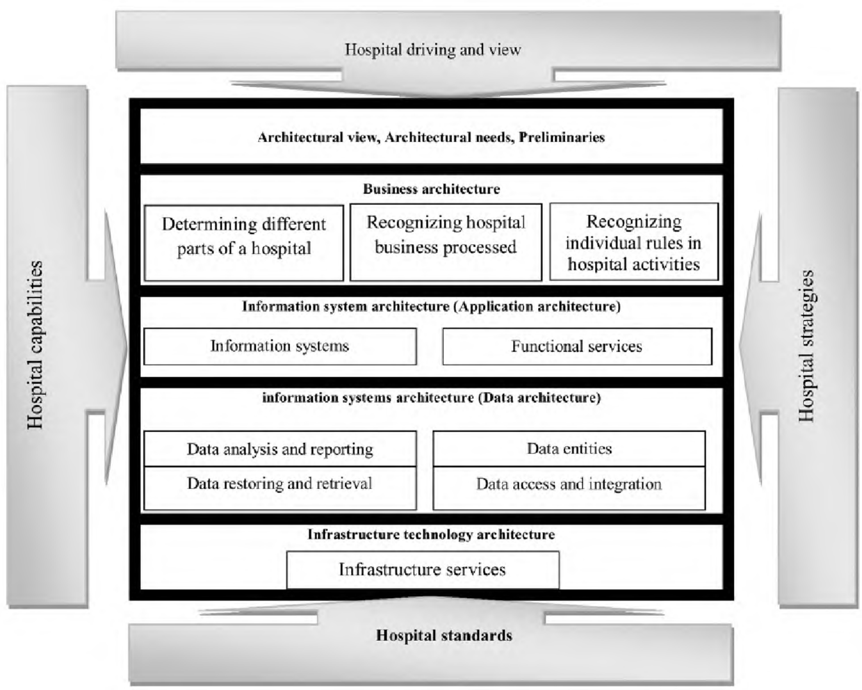

Figure 1. TOGAF model of healthcare by Haghighathoseini, et. al. (Haghighathoseini et al., 2018).

The business objects such as hospital services are not linked to data directly, but linked through behaviors known as services or business scenarios where they are usually operated in the application level. This generates an interface between the business and application layers. Data objects such as a medical record are represented at the lower technology layer as artifacts. The next section provides a methodology to consider the variables in the healthcare TOGAF model and produce an architecture that aligns the objectives of the healthcare system.

2. METHODOLOGY

Architecture evolution and selection is made using SoS Explorer, a multi-objective optimization tool utilizing a fuzzy intelligent learning architecture to assess and optimize the architecture against Key Performance Attributes (KPAs) (Curry & Dagli, 2017). The SoS Explorer was developed as part of "Flexible Intelligent Learning Architectures for System of Systems (FILA-SoS) research project of the Systems Engineering Research Center (SERC) and used on several applications (Agarwal et al., 2015; Ashiku & Dagli, 2019; Coffey & Dagli, 2019; Lesinski, Corns, & Dagli, 2016; Pape et al., 2013). The goal for using the SoS Explorer is to develop, improve, and realign the current enterprise architecture to meet ambitious strategies aligned to key capabilities and KPAs.

We perform the in the SoS Explorer the set of systems and interfaces ae defined by a vector called a chromosome by the evolutionary algorithms used to optimize the architecture. The functions **S** and **I** extract the system and interface information from a chromosome and are defined as:

$$\mathbf{S}(X,i) = \begin{Bmatrix} 1 & \text{if the } i^{th} \text{ system is selected in } X \\ 0 & \text{otherwise} \end{Bmatrix} \quad (1)$$

and

$$\mathbf{I}(X,i,j) = \begin{cases} 1 & \text{if the } i^{th} \text{ and } j^{th} \text{ systems have an interface in } X \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where X is the chromosome.

The SoS Explorer variables are:

OC	The overall capability or goal of the SoS achieved from the system-level capabilit selected systems.
C	Characteristics matrix Ns, X, Nc compo: each system and its properties represente real numbers
C'	Capability matrix Ns, X, Nc' is compose each system represented by Boolean val and its elements of functionality
Ns	Number of systems
Nc	Number of characteristics
Nc'	Number of capabilities
I	Boolean interface information between systems

2.1. IDENTIFYING SOS CAPABILITY

The individual systems such as IT systems, roles, facilities, and processes come together to meet the overall SoS capability. Each of the SoS system-level components have their own capabilities as required by the SoS and any loss of these capabilities have implications on certain KPAs. A highly capable healthcare system integrates data, workflow, and functions with the aim for high healthcare service quality, increased patient satisfaction, and reduced deaths and accidents (Figure 2). For example, when a patient is admitted in to the Emergency Room (ER), each system ensures that data (i.e. registration, medical records, etc.) are carried by various roles through various processes and IT systems to ensure the patient receives the right priority for medical attention, the right doctor, and contains information (i.e. medical history) to ensure the patient receives good treatment and a plan for exit. Therefore, each system ensures the following capabilities: data (i.e. medical records) integration, workflow (i.e. across processes and IT) integration, and functional (i.e. administration, Oncology, etc.) integration.

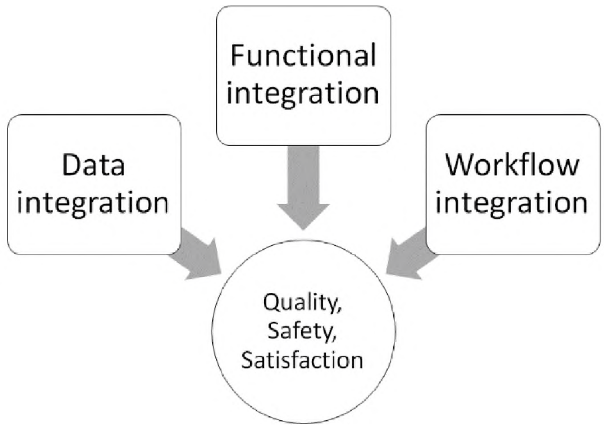

Figure 2. Healthcare SoS Capabilities.

2.2. IDENTIFYING SOS KEY PERFORMANCE ATTRIBUTES

Agility is key to success of the enterprise architecture and the key performance attributes for a healthcare SoS are:

1. Robustness: the ability to maintain effectiveness without failure and is modeled as interface redundancy:

$$\text{Robustness}(X) = -Ns + \sum_{i=1}^{Ns} \mathbf{S}(X,i) \sum_{j=1}^{Ns} \mathbf{S}(X,j)\mathbf{I}(X,i,j) \tag{3}$$

2. Cycle-time: the average performance time required to move information by a system component:

$$\text{Cycle Performance}(X,C) = \frac{\sum_{i=1}^{Ns} \mathbf{S}(X,i) C_{cycle\,performance},i}{\sum_{i=1}^{Ns} \mathbf{S}(X,i)} \tag{4}$$

3. Flexibility: the ability to employ multiple ways to succeed and the capacity to move seamless between them which is calculated by

subtracting the required capabilities from the total capabilities provided by the corresponding SoS meta architecture:

$$\text{Flexibility}(X, C') = -N_{c'} + \sum_{i=1}^{Ns} \mathbf{S}(X, i) \sum_{j=1}^{N_{c'}} C'_{ji} \quad (5)$$

4. Scalability: the ability to adapt to additions or deductions of facilities, processes and/or technologies:

$$\text{Scalability}(X, C) = \frac{\sum_{i=1}^{Ns} \mathbf{S}(X, i) C_{scalability_{,i}}}{\sum_{j=1}^{N_s} \mathbf{S}(X, i)} \quad (6)$$

5. Adaptability: the ability of a system to restructure itself in the face of business changes. This attribute is calculated using the fuzzy assessor (Pape et al., 2013) where the characteristic contributing to this attribute is the ability to be modular or restructure itself with minimal effort without disrupting the capabilities.

2.3. IDENTIFYING HEALTHCARE SYSTEMS AND CHARACTERISTICS

Haghighathoseini, et al. (Haghighathoseini et al., 2018) provided a model on an Iranian hospital which loosely identifies the systems and its characteristics in Table 1. There are a total of 20 systems and the 7 characteristics are*: cycle-time, scalability, modularity, data interoperability, benefit to patient, reusability, and decision making velocity*.

Table 1: Hospital system characteristics

	Cycle time	Scalability	Modularity	Decision Making Velocity	Benefit to patient	Reusability/Standardization	Data Interoperability
Administrative Process	1	9	8	2	1	1	1
Heath Providing Process	1	4	4	6	1	1	1
Registration Process	1	2	9	1	0	1	1
Patient Management Process	1	5	5	8	1	1	1
Facility Process	0	1	7	3	0	0	0
Disease Process	1	2	7	7	0	0	1
Public Health Warn Process	1	8	9	9	1	1	1
Doctor Selection Process	1	8	7	6	1	0	1
Intensive Care Ward	1	4	4	9	1	1	1
Hospital Clinics	1	7	6	5	1	1	1
Laboratories	1	5	7	2	1	1	1
Pharmacy	1	8	7	2	1	0	1
Headquarter Unit	1	3	2	3	1	0	1
Adminstrative units	1	7	3	4	1	1	1
Medical Doc Information System	1	8	3	8	1	1	1
Admission Information System	1	8	6	3	1	0	1
Hospital Ward Information System	1	4	4	8	1	1	1
Surgery Information System	1	4	3	8	1	1	1
Laboratory Information System	1	3	4	8	1	1	1
Pharmacy Information System	1	5	5	6	1	1	1

2.4. IDENTIFYING SYSTEM INTERFACES

The systems constituting the SoS are individual entities performing their own functions until they interface and connect with each other. The emergent behavior of the SoS is due to the coming together of individual systems and hence interfaces between the individual systems play an important role in the SoS exhibiting its capability. For the healthcare SoS, feasible interfaces between the systems are identified.

2.5. META-ARCHITECTURE GENERATION WITH FUZZY-GENETIC ALGORITHM

The purpose of the SoS Explorer application and the fuzzy-genetic algorithm is to utilize the inputs of system components and its capabilities, interfaces, and the system characteristics to generate, evaluate and optimize meta-architectures. The process of the SoS Explorer is shown in Figure 3.

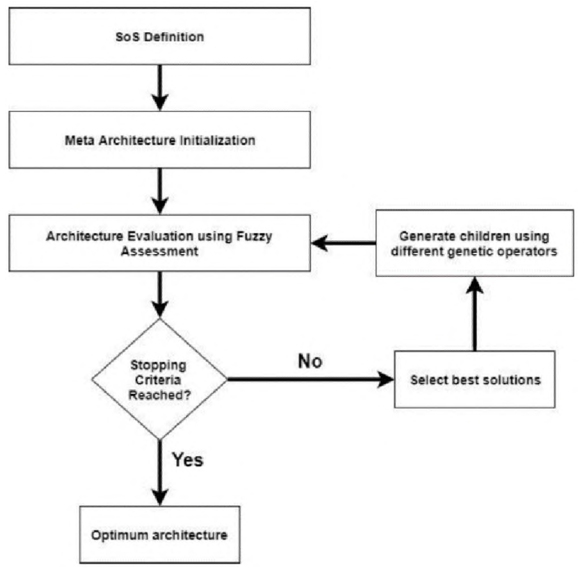

Figure 3. SoS Process Flowchart

The genetic algorithm used here is a Non-dominated Sorting Genetic Algorithm III. This algorithm searches the space of candidate architectures and generates populations of "optimal" fitness based on key performance attributes with the objective to maximize the effectiveness of the healthcare architecture. After defining the SoS, a set of chromosomes representing the meta architecture can be randomly generated with size n+(n^2-n)/2 where n is the number of systems assuming that the interfaces are bi-directional. The crisp values of the five key performance attributes are input into a fuzzy inference system (FIS) in MATLAB© and integrated into the fitness function of the genetic algorithm. The output of the algorithm is the overall KPA value of the SoS architecture based on the defined membership functions and fuzzy rules. This inference system acts as the assessment for the generated chromosomes. The best solutions from the iterations are used to generate children using different genetic operators. These chromosomes are once again evaluated using FIS. This process is repeated until the stopping criteria is reached, which is the number of iterations for the genetic algorithm and the best solution will be the final meta architecture for the SoS.

3. RESULTS

The SoS Explorer produced an optimal meta-architecture, as a result of the genetic algorithm optimization, shown in Figure 4 with results of KPA values and overall score shown in Table 2. The systems in the meta-architecture shown in the filled circles represent active systems and the lines between circles are bi-directional interfaces. All but two systems (disease and facility processes) participated with many of the systems having at least four interfaces. The reason for not including the disease and facility processes is their inability to integrate data and workflow in the architecture. However, the IT systems are utilized to manually manage data for disease and facilities and are integrated into the overall architecture, but there is an opportunity to explore ways to facilitate the integration with automation.

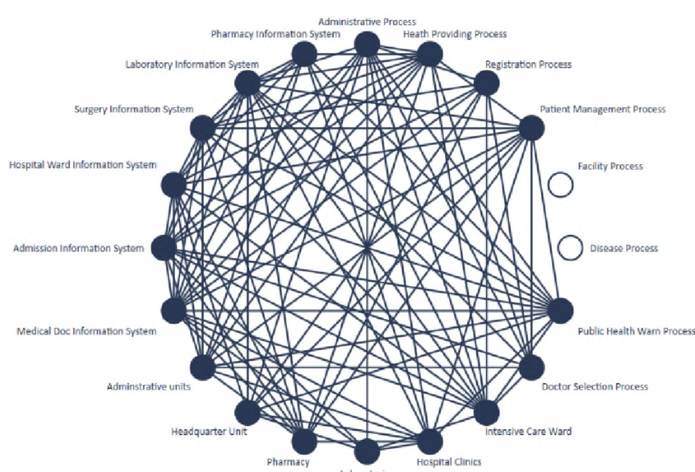

Figure 4. Healthcare SoS Meta-Architecture

Table 2. Meta-Architecture Results

Algorithm	NSGA-III
Division	3
Probability of mutation	0.005
Probability of crossover	1
Optimal Architecture	
Scalability	100
Robustness	100
Flexibility	97.5
Cycle Performance	58.02
Overall	50

4. CONCLUSION AND FUTURE WORK

The design and assessment of an enterprise architecture is a complex and extremely expensive task. This paper offers an affordable example of using computational intelligence approaches to assess a common, yet complex enterprise architecture of the healthcare system with the objective to provide high healthcare service quality, increased patient satisfaction, and reduced deaths and accidents. To meet the objective is to have well-defined system interfaces that drive interoperable healthcare processes, services, and systems where agility is a key attribute. The SoS Explorer application was used with fuzzy logic, genetic algorithms, and mathematical programming to generate, assess, and optimize meta-architectures against key performance attributes of agility.

The healthcare meta-architecture produced in this study is not a solution to address the challenges of the healthcare enterprise architecture but provides insight on the areas – systems, capabilities, characteristics, and particularly the interfaces – to pay attention to where interfaces have implications on agility, an important attribute and not to be severely compromised. The results provide possibilities for future work such as exploring accurate mathematical modeling to best fit the problem scheme and evaluating the validity of the meta-architecture model for real world heathcare systems. There is a need to apply SoS with precise healthcare system data to understand which of the interfaces and the system have the greatest implications to performance.

REFERENCES

Agarwal, S., Pape, L. E., Dagli, C. H., Ergin, N. K., Enke, D., Gosavi, A., Gottapu, R. D. (2015). Flexible and intelligent learning architectures for SoS (FILA-SoS): Architectural evolution in systems-of-systems. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2015.03.005

Alberts, D. S., & Hayes, R. E. (2006). Understanding Command and Control. *The Journal of Military History*. https://doi.org/10.1353/jmh.2007.0051

Ashiku, L., & Dagli, C. (2019). Cybersecurity as a centralized directed system of systems using SoS explorer as a tool. *2019 14th Annual Conference System of Systems Engineering, SoSE 2019*. https://doi.org/10.1109/SYSOSE.2019.8753872

Bredemeyer, D.; Krishnan, R.; Lafrenz, A.; Malan, R. (n.d.). Enterprise Architecture as Business Architecture. Retrieved January 1, 2020, from 2003 website: http://www.ewita.com/newsletters/10025_files/EnterpriseArchitectureAsCapabilitiesArchSlides.PDF

Christopher, M., & Towill, D. R. (2000). Supply chain migration from lean and functional to agile and customised. *Supply Chain Management*. https://doi.org/10.1108/13598540010347334

Coffey, G. P., & Dagli, C. (2019). A Method to Use the SoS Explorer Application with Fuzzy-Genetic Algorithms to Support Military Veterans within Higher Education. In *Systems Engineering in Context*. https://doi.org/10.1007/978-3-030-00114-8_19

Conrow, E. H. (2011b). Estimating Technology Readiness Level Coefficients. *Journal of Spacecraft and Rockets*. https://doi.org/10.2514/1.46753

Curry, D. M., & Dagli, C. H. (2017). SoS explorer: A tool for system-of-systems architecting. In *Disciplinary Convergence in Systems Engineering Research*. https://doi.org/10.1007/978-3-319-62217-0_14

Dooley, K. (1997). A Complex Adaptive Systems Model of Organization Change. *Nonlinear Dynamics, Psychology, and Life Sciences*. https://doi.org/10.1023/A:1022375910940

Gaona Caceres, J. L., & Rosado Gómez, A. A. (2019). Business architecture: A differentiating element in the growth of organizations. *Journal of Physics: Conference Series*. https://doi.org/10.1088/1742-6596/1257/1/012007

Haghighathoseini, A., Bobarshad, H., Saghafi, F., Rezaei, M. S., & Bagherzadeh, N. (2018). Hospital enterprise Architecture Framework (Study of Iranian University Hospital Organization). *International Journal of Medical Informatics*. https://doi.org/10.1016/j.ijmedinf.2018.03.009

Harishankar, R., & Daley, S. K. (2011). Actionable business architecture. *Proceedings - 13th IEEE International Conference on Commerce and Enterprise Computing, CEC 2011*. https://doi.org/10.1109/CEC.2011.54

Hazen, B. T., Bradley, R. V., Bell, J. E., In, J., & Byrd, T. A. (2017). Enterprise architecture: A competence-based approach to achieving agility and firm performance. *International Journal of Production Economics*. https://doi.org/10.1016/j.ijpe.2017.08.022

Herzlinger, R. E. (2006). Why innovation in health care is so hard. *Harvard Business Review*.

Kotusev, S. (2018). TOGAF-based enterprise architecture practice: An exploratory case study. *Communications of the Association for Information Systems*. https://doi.org/10.17705/1CAIS.04320

Lesinski, G., Corns, S. M., & Dagli, C. H. (2016). A fuzzy genetic algorithm approach to generate and assess meta-architectures for non-line of site fires battlefield capability. *2016 IEEE Congress on Evolutionary Computation, CEC 2016*. https://doi.org/10.1109/CEC.2016.7744085

Macrae, C., & Stewart, K. (2019). Can we import improvements from industry to healthcare? *BMJ (Online)*. https://doi.org/10.1136/bmj.l1039

Malta, P., & Sousa, R. D. (2016). Process Oriented Approaches in Enterprise Architecture for Business-IT Alignment. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2016.09.239

McDaniel, R. R., Lanham, H. J., & Anderson, R. A. (2009). Implications of complex adaptive systems theory for the design of research on health care organizations. *Health Care Management Review*. https://doi.org/10.1097/HMR.0b013e31819c8b38

Olsen, D. H. (2017). Enterprise Architecture management challenges in the Norwegian health sector. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2017.11.084

Pape, L., Giammarco, K., Colombi, J., Dagli, C., Kilicay-Ergin, N., & Rebovich, G. (2013). A fuzzy evaluation method for system of systems meta-architectures. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2013.01.026

Ramasesh, R., Kulkarni, S., & Jayakumar, M. (2001). Agility in manufacturing systems: An exploratory modeling framework and simulation. *Integrated Manufacturing Systems*. https://doi.org/10.1108/eum0000000006236

Rouse, W. (2008). Health care as a complex adaptive system: implications for design and management. *Bridge-Washington-National Academy of Engineering-*.

Vincent, C., & Amalberti, R. (2016). Safer healthcare: Strategies for the real world. In *Safer Healthcare: Strategies for the Real World*. https://doi.org/10.1007/978-3-319-25559-0

Wickramasinghe, N., Chalasani, S., Boppana, R. V., & Madni, A. M. (2007). Healthcare system of systems. *2007 IEEE International Conference on System of Systems Engineering, SOSE*. https://doi.org/10.1109/SYSOSE.2007.4304283

II. A CLUSTER-BASED FRAMEWORK FOR INTERFACE ANALYSIS IN LARGE-SCALE AEROSPACE SYSTEMS

Josh Goldschmid* and Steven M. Corns

Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, USA

*jhgrm5@umsystem.edu

Deparment of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409

ABSTRACT

This paper proposes a framework using a community maturity level metric to determine the integration readiness of interface elements in a particular network cluster. As a proof of concept this methodology is applied on an aircraft seating system to assess the readiness of complex interfaces before proceeding to full-scale production and systems integration. A multi-objective genetic algorithm, MOGA-Net, is coupled with the Newman-Girvan modularity metric as a clustering algorithm. This algorithm identifies system elements grouped by common interfaces, referred to as community clusters. The TRL and IRL values for these elements is then used to calculate an overall community maturity level. The achieved performance in these clusters is then compared to the target performance to determine overall maturity of the interfaces. This is compared to other system readiness metrics and interface readiness metrics as applied to the aircraft seating system and was found to be more consistent with subject matter expert evaluations during the Critical Design Review. This gives a better representation of the

true readiness of system interfaces before entering design reviews to reduce overall integration risks

1. INTRODUCTION

Interface analysis is a difficult systems engineering task that if not conducted sufficiently early in the development lifecycle will lead to systems integration that occurs late in the development process is at considerable risk of failure with severe cost and schedule consequences (GAO, 2018). In addition, the likelihood of difficulties during the integration phase increases as interfaces grow in scale and complexity. For example, the F-16 from the 1970s had 15 subsystems, 10^3 interfaces and less than 40% of its functions were managed by software while the F-35 has 130 subsystems with approximately 10^5 interfaces and over 90% of its functions are managed by software (Arena et al., 2008), increasing the complexity of verification activities and integration risks. Because of this, many aircraft manufacturers have established an interface control working group (ICWG) to bring together stakeholders to identify and track interfaces.

This paper proposes a clustering-based interface assessment framework (CIAF) to address some of the interface analysis challenges in a large-scale system development lifecycle such as the ones often experienced in aircraft development (Kapurch, 2010). The CIAF proposed in this paper uses a seat power system example as part of a commercial 777 aircraft to demonstrate the framework's effectiveness, comparing results to the current interface analysis methods used during critical design review (CDR). The CIAF uses a metric called “Community Maturity Level” (CML) to determine the technical maturity for a particular "community" of interface elements. Using clustering techniques, the power seat subsystem is decomposed into communities of components with dense interfaces and using the CML metric with performance analysis, the readiness of these proposed communities for system development and demonstration are determined. The input to the CIAF is a Design Structure Matrix (DSM), where in the seat example at CDR the DSM has 35 hardware and software components. The physical architecture is determined by part number identification and therefore the 35 components in the DSM have unique part identification. The authors believe the CIAF can successfully perform analysis using DSM input with functional and logical components of the system architecture.

2. BACKGROUND

There are many approaches to assess complex interfaces and interactions such as design structure matrix and technology maturity assessment. All of these approaches support the development of the CIAF to enable interface analysis.

2.1. DESIGN STRUCTURE MATRIX

The Design Structure Matrix (DSM) is a representation and analysis tool that models interface elements (Eppinger & Browning, 2012) such as physical proximity, functions, and environment. The DSM is a square matrix, akin to the traditional N^2 chart and the SV-3 in the DoD Architecture Framework (DODAF) (DoD, 2010) allowing systems engineers to investigate coupling between components of the system. While the

DSM provides a compact way of representing a system and its interfaces, it does not capture the multipartite relationships found in complicated aircraft systems with dense interconnections and sparse intraconnections.

2.2. MODEL-BASED SYSTEMS ENGINEERING

Model-based Systems Engineering (MBSE) (Department of Defense, 2008), a design process-agnostic approach that uses single-source-of-truth models to convey the system design, perform analysis, and assure maturity throughout the development process and entire system life-cycle, has been delivering value in the development of complex systems (Burkhart, Friedenthal, Griego, Sampson, & Spiby, 2007; Madni & Sievers, 2018). This value is realized when technical, cost, and schedule risks are mitigated throughout the development process (Estefan, 2007) by helping engineers automate and facilitate requirements traceability and manage interfaces in models. These models are used to support trade studies, change impact analysis, and verification & validation (V&V) activities (Corns & Gibson, 2012; Friedenthal, Moore, & Steiner, 2014; Long & Zane, 2011; Oliver, Kelliher, & Keegan, 2004). Though MBSE enables assessment of interfaces and interactions with a useful set of modeling constructs that capture complex structural, behavioral, and requirements relationships in a system, interface analysis can be extremely challenging with models for large-scale systems containing gigabytes of data (Carson, 2015; Malone, Friedland, Herrold, & Fogarty, 2016; Voirin, Bonnet, Normand, & Exertier, 2015). Model-Based Systems Engineers are encouraged to use the CIAF which provides them network science and interface analysis techniques and visualizations that help gain insight on the interfaces and its maturity, build intuition, improve stakeholder collaboration, and improve the fidelity of their architectural models.

2.3. TECHNOLOGY MATURITY ASSESSMENT

The technology maturity assessment (TMA) is an assessment technique proposed by Bilbro that uses a work breakdown structure (WBS) to identify key technologies subject to the technology readiness level (TRL) maturity scale (Bilbro, 2007). The TRL scales, originally developed by National Aeronautics and Space Administration (NASA), range from 1 to 9, indicating increasing maturity and technology risks. Bilbro argues that the maturity of a TRL 9 technology drops to TRL 5 when it is integrated into a new environment or configuration. For most large and complex systems there are too many WBS elements to address and track individually and so a metric is needed to capture readiness of these systems.

To address the limitations of TRLs, (B. J. Sauser, Marquez, Henry, & DiMarzio, 2008) proposed the System Readiness Level (SRL) metric to provide a system-level view in real-time of the system development and maturity in relation to the Department of Defense's (DoD) Phases of Development, giving managers opportunities to take proactive measures to reduce developmental risk. The SRL metric introduced Integration Readiness Level (IRL) variables to determine the overall SRL. The IRL is a measurement of the interface compatibility indicating maturity between interface elements (B. Sauser, Gove, Forbes, & Ramirez-Marquez, 2010). IRLs scale similarly to TRLs but start with zero instead of one, with zero indicating there is no interface (e.g. A and D & B and D in Equation 1) or no integration has been planned or intended. Figure 1 shows an example of a system under consideration (real system examples found in S. Yasseri & Bahai, 2018; S. F. Yasseri & Bahai, 2020), where the TRL for each of the subsystems is identified as A, B, C, and D.

$$SRL = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \overset{\text{Normalized IRL Matrix}}{\begin{array}{cccc} A & B & C & D \\ \end{array}} \begin{bmatrix} 1 & {}^{1}/_{9} & {}^{3}/_{9} & 0 \\ {}^{1}/_{9} & 1 & {}^{6}/_{9} & 0 \\ {}^{3}/_{9} & {}^{6}/_{9} & 1 & {}^{7}/_{9} \\ 0 & 0 & {}^{7}/_{9} & 1 \end{bmatrix} \times \overset{\text{TRL Matrix}}{\begin{pmatrix} {}^{3}/_{9} \\ {}^{6}/_{9} \\ {}^{7}/_{9} \\ {}^{9}/_{9} \end{pmatrix}} = 0.49 \qquad (1)$$

The IRL represents the maturity of the interfaces between these subsystems. These subsystems are represented in Equation 1 with the normalized IRL matrix capturing the interface maturity and the TRL capturing the subsystem maturity. Calculating the overall SRL for this sample system, it can be seen that the SRL is lower than all but one TRL. This 0.49 SRL value shows that even though most of the components are ready for the System Development or Production Development phase (Magnaye, Sauser, & Ramirez-Marquez, 2010), the system taken as a whole should still be in the technical development phase.

The ordinal scales of TRLs and IRLs do not convey information about the degree of differences between measures, making arithmetic calculations with these scales of limited utility (Conrow, 2011a; Kujawski, 2013; McConkie, Mazzuchi, Sarkani, & Marchette, 2013). In addition, it is important to evaluate the connectivity or key functional thread of important interfaces against performance requirements, such that a missing or immature interface can have performance implications on the system architecture. A method for determining communities comprised of highly networked interfaces is needed to bring analytical focus on the system's integration readiness.

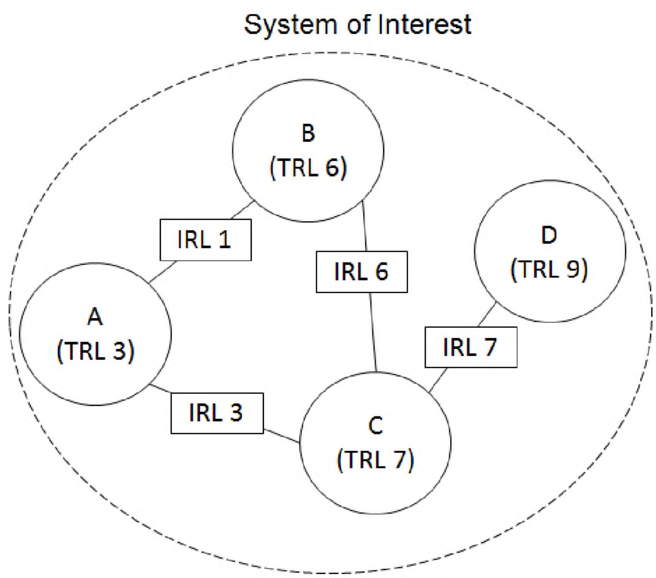

Figure 1. System of Interest Readiness Level

2.4. COMMUNITY DETECTION ALGORITHMS

Clustering is one method used to organize the interface data into meaningful structures, or communities. These communities can provide topographical insights on a complex network's underlying hidden attributes. Figure 2 gives a example of how a network may be partitioned using graph-based clustering algorithms.

Several researchers have used clustering based community-detection algorithms to analyze community structures in complex systems Lancichinetti & Fortunato (2009) performed a comprehensive assessment of community detection algorithms. Tamaskar, Neema, & Delaurentis (2014) developed a framework for measuring complexity of aerospace systems based on size, coupling, and modularity using the Newman-Girvan algorithm to decompose the system into modules. Dabkowski, Valerdi, & Farr (2014) treated the DoDAF Systems View 3 (SV-3) as an adjacency matrix and used the Newman-Girvan community detection heuristic to divide the SV-3 into groups of subsystems such that the number of interfaces are dense within and sparse between groups. Pizzuti (2012) developed a multi-objective genetic algorithm (MOGA-NET) to uncover community structures in complex networks.

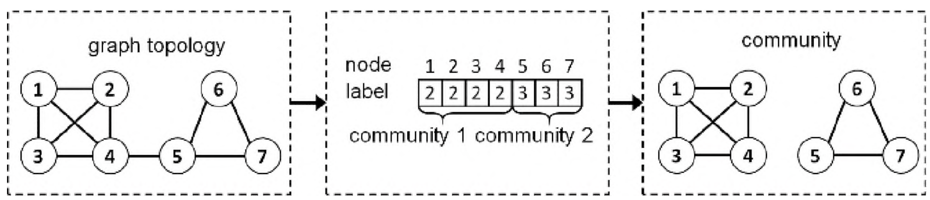

Figure 2. Illustration of clustered communities.

The assessment tools available are useful. The CIAF synthesizes these tools and offers a framework that

- Includes an assessment of system integration readiness metrics that several industries use;
- Provides systems engineers with a method to identify related interfaces (communities) in the architecture, exposing missing or immature interfaces;
- Assists in the identification and quantification of measures of performance associated to interface issues at all levels in the system.

3. FRAMEWORK DESCRIPTION

The CIAF is a two-stage process resulting in a thorough integration readiness analysis of complex interfaces in a system (Figure 3). The first stage establishes the communities using Pizzuti's evolutionary-based clustering algorithm (MOGA-Net) (Pizzuti, 2012) to identify a solution set of communities from the Design Structure Matrix (DSM) of interface components and a solution is selected using Newman-Girvan's modularity metric (Newman, 2004a). In the the second stage of the framework, each of the communities from the selected solution is assessed to determine the maturity level of the interfaces and performance measures within the communities.

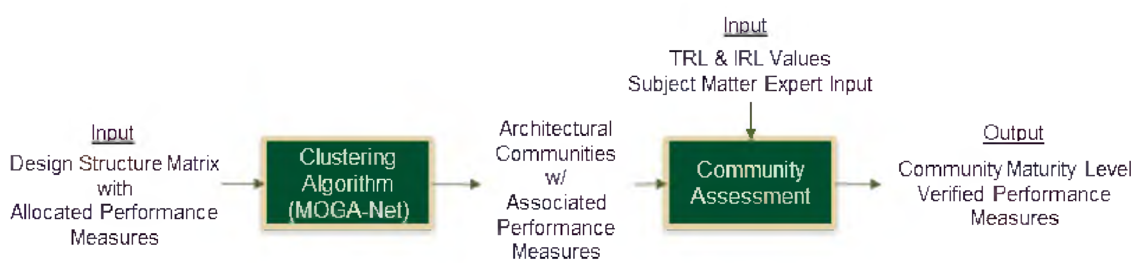

Figure 3. Two-Stage Clustering-based Interface Assessment Framework.

3.1. COMMUNITY DETECTION ALGORITHM

Communities are identified through clustering using Pizzuti's MOGA-Net (multi-objective genetic algorithm) which applies a genetic algorithm (GA) (Mitchell, 1998) to identify communities, providing a set of solutions contained in the Pareto front (Pizzuti, 2012). This approach was selected because it has been shown to identify clusters within complex engineering systems. Each of these solutions corresponds to a trade-off between two objective functions, the community score (CS) (Pizzuti, 2012) that measures the density of clusters obtained and community fitness, $P(V_i)$ (Lancichinetti & Fortunato, 2009), maximizes the internal degrees of the interface nodes within a community and minimizes the external links between communities.

To select a solution contained in the Pareto front, the MOGA-Net adopts the Girvan and Newman modularity metric (Q) (Newman, 2004a), providing a solution with generally the highest modularity value.

The community score (CS) is one input for the MOGA-Net, and is represented as

$$CS = \sum_{i=1}^{m} M(V_i) \cdot uv_i \qquad (2)$$

where:

- uv_i is the volume of community V_i, i.e. number of links (edges) connecting the components or nodes (vertices) in V_i.
- $M(V_i)$ is the power mean of V_i of order r.

The power mean defines the fraction of interconnections among nodes and is defined as:

$$M(V_i) = \frac{1}{|V_i|} \sum_{j \in V_i}^{m} \left(\frac{k_j^{in}}{|V_i|} \right)^r \qquad (3)$$

where k_j^{in} represents the number of j node connections in community V_i. The higher the community score, the denser the community is obtained. The community fitness function is defined as:

$$P(V_i) = \sum_j \frac{k_j^{in}}{\left(k_j^{in} + k_j^{out}\right)^{\alpha}} \quad (4)$$

where k_j^{out} represents the number of j node connections external of community V_i and α is a resolution parameter controlling the scope of the community. If you increase alpha, you obtain smaller communities (Figure 4) which can influence the community fitness which is based on the sum of the fitness of nodes belonging to a cluster. When the sum reaches its maximum value, the number of external links is minimized.

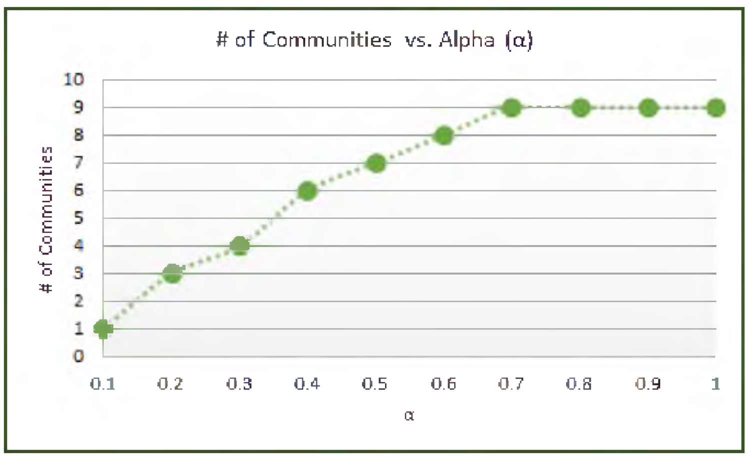

Figure 4. Alpha Controlling the Scope of Communities.

3.1.1. Multi-Objective Genetic Algorithm. Pizzuti's MOGA-Net (multi-objective genetic algorithm) uses a DSM to represent the system as a graph S = (E,V) where V ={v1, v2, ...,vn} is a set of components in the system, and E = {e1, e2, ...,en} is an interface between components (Figure 5).

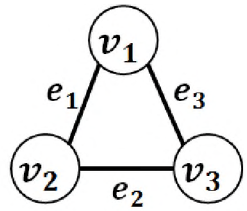

Figure 5. Graph $S = (E, V)$

A group of components of high density of interfaces within them and low density between groups in the network forms a community or cluster. The matrix representing S captures whether or not there is an interface between components i and j where the interface can represent spatial, energy, material, or information interactions:

$$S_{ij} = \begin{cases} 1 & \text{if an interface exists between nodes i and j} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The CIAF does not differentiate between interface types nor between components and subsystem definitions when representing the system in the GA. The CIAF allows the user to determine the level of abstraction in the DSM and use the algorithm to identify the clusters that are then examined to reduce any interface risks. This is done by comparing the community maturity level, described in the next section, of the components that share the interface and interface readiness level for that interface. The MOGA provides a set of solutions, but provides little information on the strength of the community structure. To identify the strongest community structure, the modularity metric is used.

The modularity metric identifies which solution on the Pareto Front should be selected to determine the community maturity level. The Newman-Girvan modularity metric (Q) is represented in Equation 6:

$$Q = \sum_{k=1}^{s} \left[\frac{l_k}{m} - \left(\frac{d_k}{2m} \right)^2 \right] \tag{6}$$

where m is the sum of links in the network; l_k is the number internal edges (links) of a community k; and d_k is the sum of the degrees of all vertices in the community k.

The modularity measure is calculated each time the network is divided into communities. The process of removing links, clustering components into communities, and calculating modularity continues until Q reaches a local maxima where the value of Q approaches 1, signaling a strong community structure and therefore, the solution. The solution identified with a set of communities is then analyzed to determine the maturity of the interacting components within each community.

3.2. COMMUNITY MATURITY LEVEL

In the assessment of a system, Kujawski (2013) suggested models to bring transparency to the system's readiness: the tabular model and the system's Min TRL-Min IRL model. The tabular model that summarizes the example system of interest (Figure 1) is shown in Table 1, with the count of each level of TRLs and IRLs. As described by Kujawski, the system's Min TRL-Min IRL model indicates that from a risk perspective the overall maturity is based on minimum TRL and IRL values, in contrast to the SRL measure. The *min*(TRL), *min*(IRL) = (3,1) according to Table 1 implies that the system has low readiness level when compared to the SRL of 0.49. The authors propose to use the Min TRL-Min IRL model approach to evaluate the integration maturity of each community in a system defined by the clustering algorithm.

Table 1. Tabular SRL Model

Level	1	2	3	4	5	6	7	8	9	10
TRL_i			1			1	1		1	
IRL_i	1		1			1	1			

The "Community Maturity Level" (CML) metric determines the technical maturity of each community of components with dense interfaces using the Min TRL-Min IRL model approach (Figure 6). The CML analysis is then used with the assessment of performance measures within the community to provide information on integration readiness. This means the integration risk perspective is not driven solely by the lowest IRL and TRL in a given community, but also the performance of interface components to understand the overall integration implications.

The community maturity level is calculated using Equation 7:

$$CML(u, v) = \sqrt{(u * v)}\ /100 \quad (7)$$

Where u = minimum TRL and v = minimum IRL.

Table 2 provides the CML values based on Equation 7 for all minimum TRL and minimum IRL values. The table serves as a guide from a risk standpoint, in comparison to using the SRL metric, to assess with other methodologies the maturity of technology and its interfaces within a community.

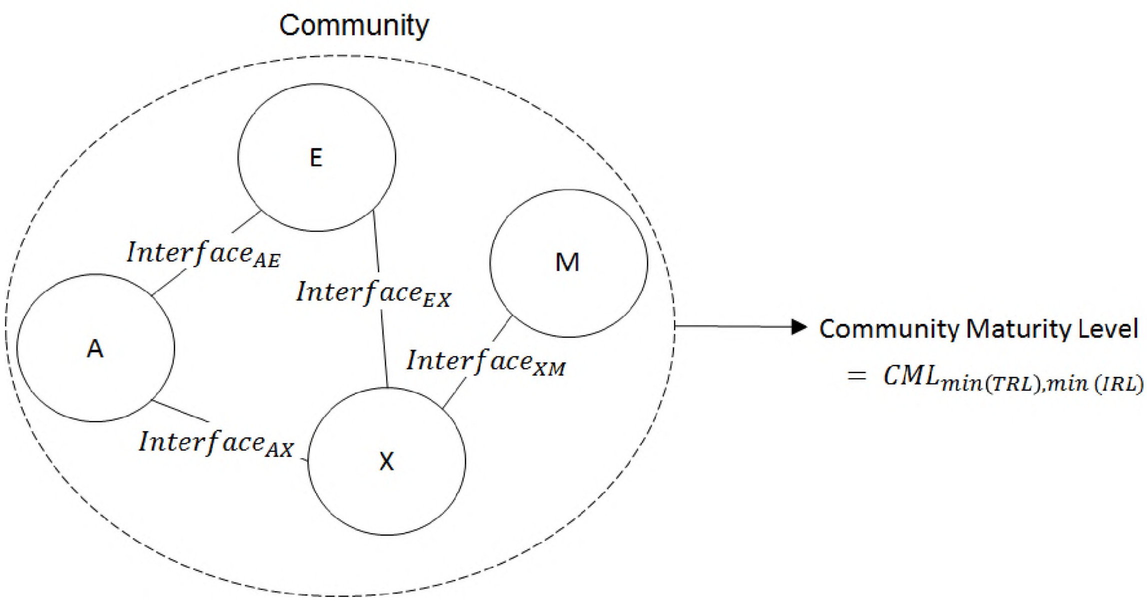

Figure 6. Community Maturity Level

Table 2. Community Maturity Level Metric calculated from min TRL, min IRL

		min IRL								
		1	2	3	4	5	6	7	8	9
min TRL	1	0.10								
	2	0.14	0.20							
	3	0.17	0.24	0.30						
	4	0.20	0.28	0.35	0.40					
	5	0.22	0.32	0.39	0.45	0.50				
	6	0.24	0.35	0.42	0.49	0.55	0.60			
	7	0.26	0.37	0.46	0.53	0.59	0.65	0.70		
	8	0.28	0.40	0.49	0.57	0.63	0.69	0.75	0.80	
	9	0.30	0.42	0.52	0.60	0.67	0.73	0.79	0.85	0.90

The scale and definitions of the various levels of the CML proposed here are adapted from Sauser's (2008) research and are correlated to phases of the systems engineering life-cycle (Figure 7) . It is important to note that a community that has not reached full maturity is capable of transitioning into the production phase, with the caveat the key performance measures associated to the community demonstrates with a certain confidence level (acceptable risk) that the technical performance measure targets are on track to be met.

CML	Phase	Definitions
0.90 to 1.00	**Operations & Support**	Execute a support program that meets material readiness and operational support performance requirements and sustains the community in the most cost-effectice manner over its total lifecycle
0.80 to 0.89	**Production**	Achieve operational capability that satisfies mission needs
0.50 to 0.79	**Engineering & Manufacturing Development**	Develop capability of clustered, interfacing technologies ; reduce integration and manufacturing risk; ensure operational supportability of the cluster; minimize logistics footprint; implement human systems integration; design for production; ensure affordability and protection of critical program informationl and demonstrate community integration, interoperability, safety, and utility
0.20 to 0.49	**Technology Development**	Reduce technology risks and determine appropriate set of technologies to integrate to serve key functions
0.10 to 0.19	**Concept Refinement**	Refine initial concept. Develop technology and interface strategy

Figure 7. Community Maturity Level Assessment with definitions and the associated phase of system development

Using Table 1 as an example community in a system where the minimum TRL is 3 and the minimum IRL is 1, the equation is as follows:

$$CML_{minTRL,minIRL} = \sqrt{(3 * 1)/100} = 0.17 \tag{8}$$

For example, based on Figure 7, the result of equation 8 indicates that the community is in the concept refinement phase. In contrast, if the community is assumed to be a system as a whole, the SRL value would be 0.49, indicating it is past the concept phase in the technology development phase. If the system described were to reach Critical Design Review (CDR) where we review the system and freeze the detailed design, the community maturity level of (3,1) = 0.17 (Equation 8) presents potential integration risks and the technology and interfaces would need to be examined prior to integration.

After the communities are scored, the CIAF evaluates the readiness between communities to determine how ready they are for integration into the system, using the same principle when assessing individual communities. Figure 8 presents an example of how some communities are connected via interfaces between components. Communities are surrounded by lines, and any component that is contained within more than one community is an interface between those communities.

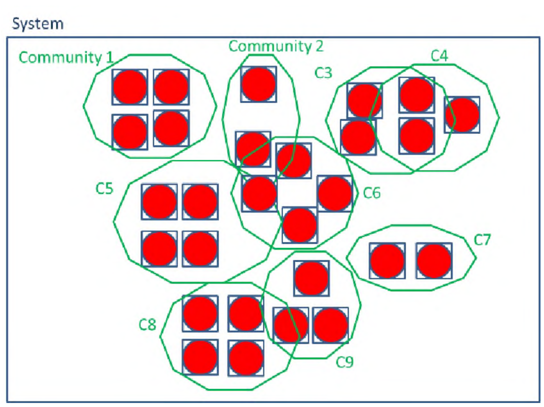

Figure 8. Integration of Communities

3.3. PERFORMANCE MEASURES AND COMMUNITIES

Using the CML alone to assess interfaces and integration readiness is not sufficient to mitigate overall integration risk. The framework adds another layer of analysis by cross-examining the CML analysis with the analysis of performance measures associated to each cluster to validate the community maturity level. The NASA Systems Engineering Handbook describes the relationships between performance measures (NASA, 2016). The qualitative, mission-based Measure of Effectiveness (MOE) are decomposed into Measure of Performance (MOP) and Technical Performance Measures (TPMs) that provide qualitative and quantitative "design to requirements" measures.

TPMs quantitatively measure the attributes of a system element within the system to determine its compliance, at a given time, to key requirements. TPMs are measured against the expected requirement or threshold at a given time. The achieved performance at time *i*, AP_{ij} is the percentage of the TPM threshold for component *j*, relative to its measured performance, MP_{ij}, up to the value of 1.0. Any value over 1.0 indicates the performance of the component has exceeded the required performance.

$$AP_{ij} = \left\{ \frac{\text{TPM Threshold}}{MP_{ij}}, 1 \right\} \tag{9}$$

Insight regarding confidence of achieving MOEs are provided by MOPs at the system-level of the technical solution, which are traceable to lower-level parameters measured by TPMs through the requirements allocation process. Thus, before inputting the DSM into the clustering algorithm, it is important to have a comprehensive set of key measures allocated to the elements in the system (Figure 9) which is represented by the DSM. In some cases, satisfying a TPM involve multiple system elements. In other words, when TPM allocations are in place, a clear and traceable method should be defined.

When the DSM is exploited into architectural communities as a result of the clustering algorithm. This associates the performance measures to the communities to verify the cluster found using the Newman-Girvan's Q modularity metric. One can assess for interface gaps based on the Community Maturity Level and how each community will achieve its higher level performance measure as the components and subsystems are integrated.

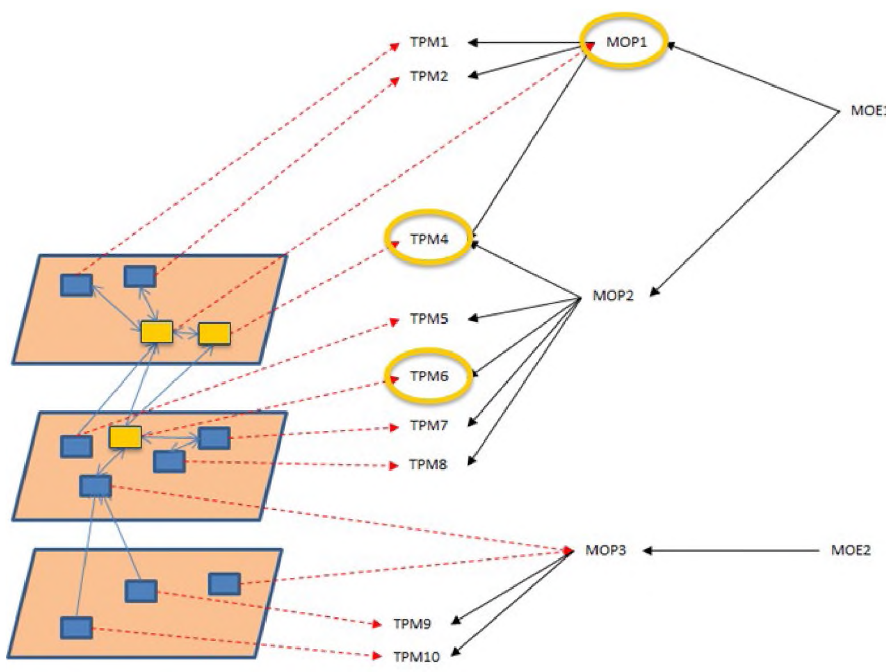

Figure 9. Allocation of Measures to System Elements. The yellow boxes indicate these components are placed into a community. The performance measures associated to the highlighted components are clustered as a result of the clustering process.

4. FRAMEWORK DESCRIPTION

To show the efficacy of the CIAF, an evaluation of the framework was performed using Critical Design Review (CDR) results of the powered seat system in a commercial aircraft (Figure 10). The CIAF validates that the community network structure is not always aligned to the seat hierarchical structures as defined by engineers in various specialties. In other words, the community detection algorithm may include components in a community for interface analysis that is not found in the hierarchical structure. There may be specific interfaces related to a seat system that are difficult to identify and verify for an engineer who works in a different specialty. By including components in the network that is not intuitive to a specialized engineer, the CIAF can provide warning of vulnerabilities to microscopic disturbances in complex systems (Carlson & Doyle, 2002). These disturbances are usually manifested during late systems verification phases.

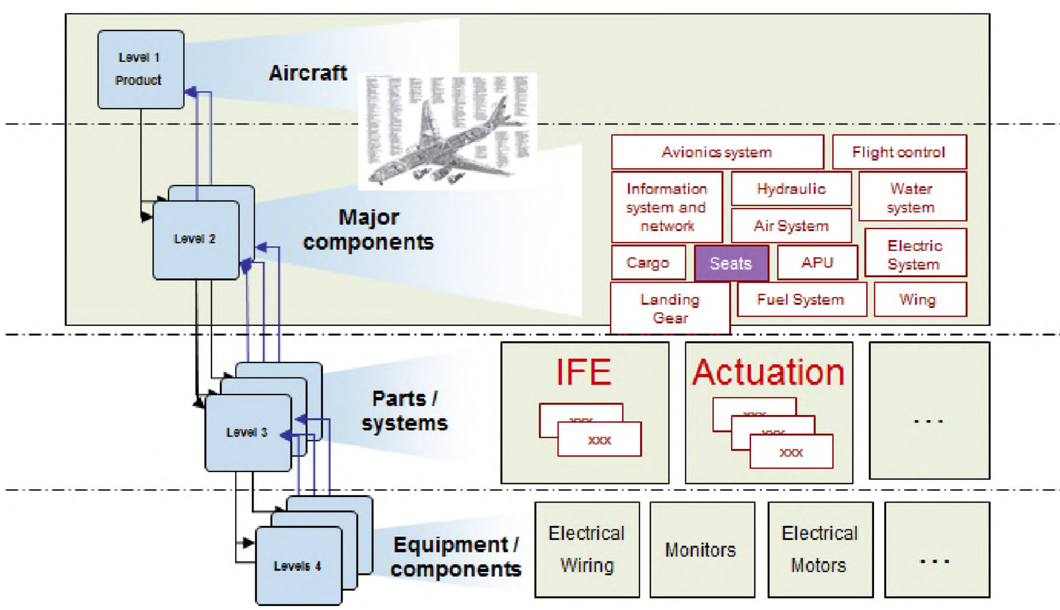

Figure 10. Aircraft Seat System Decomposition.

4.1. STEP 1: MODEL SEAT SYSTEM INTERFACES IN DSM FORMAT

The design structure matrix (DSM) representing a simplistic view of seat system at the CDR is a symmetric matrix of 35 rows and 35 columns where blue square entries in Figure 11 represent interfaces between the components.

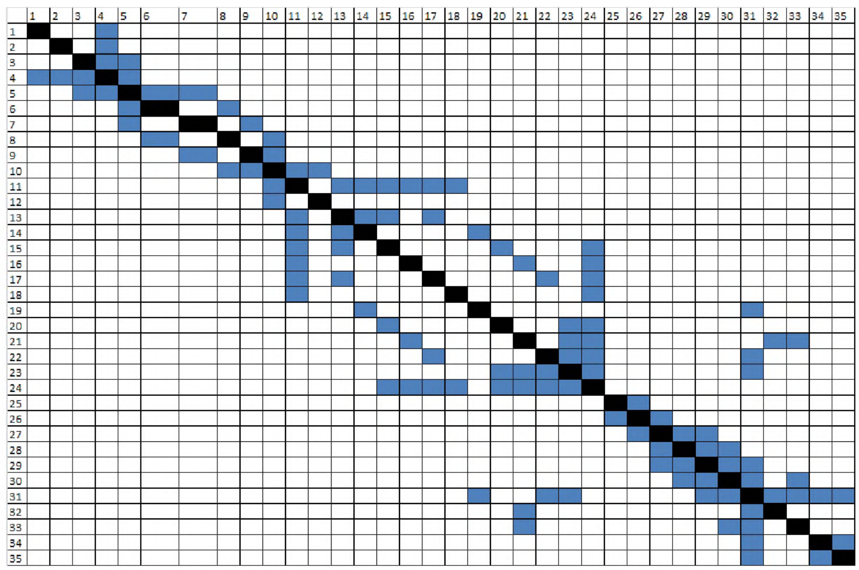

Figure 11. Design Structure Matrix of seat system and airplane interfaces

4.2. STEP 2: ALLOCATE PERFORMANCE MEASURES TO SYSTEM ELEMENTS

The seat system performance measures were allocated to the structural elements in the system and organized in the matrix format (Figure 12). This performance allocation is an expected system engineering activity that prepares us for the next step of which performance measures in the cluster needs to be analyzed with the CML. MOPs and MOEs are identified and traced to TPMs (NASA, 2016). Some system elements and interfaces may not have quantitative TPMs or MOPs allocated, but are directly allocated to MOEs due to its significant customer-level value (Figure 13). The TPMs are marked "X" where the 'X' values in the TPM section indicate the achieved performance, AP_{ij}, as calculated in Equation 9. The MOP "X" indicates its association to a component or a TPM within the community and can be calculated using Equation 9. MOEs marked as "X" in a community indicates that an associated MOP falls in the community. If the MOE is quantitative, then it can be calculated using Equation 9. Otherwise, it can be expressed as probabilities that the system will perform as required.

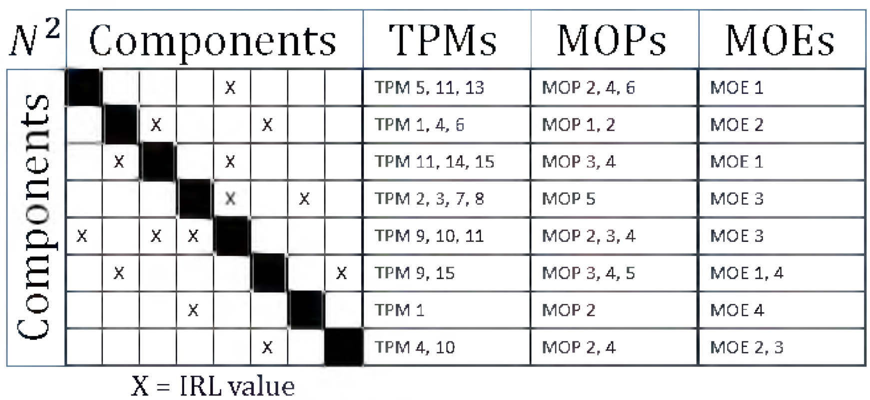

Figure 12. Example format of DSM and allocated performance measures for each interface

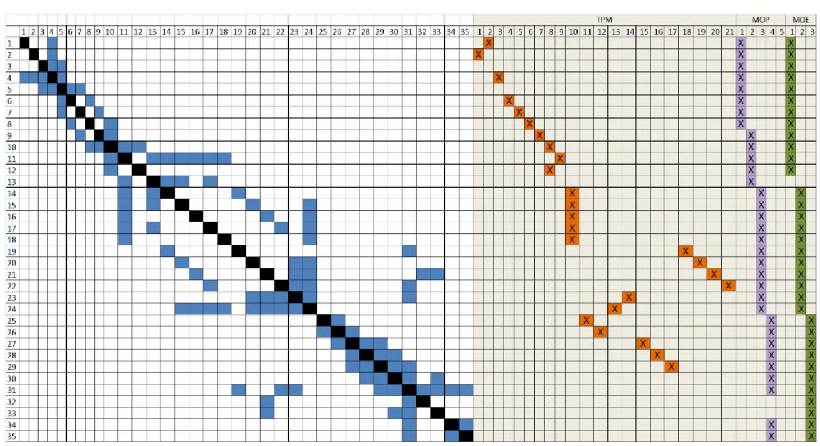

Figure 13. Snapshot of aircraft seat system with allocated performance measures using Figure 12 format

4.3. STEP 3: EXPLOIT ARCHITECTURAL COMMUNITIES AND CLUSTER PERFORMANCE MEASURES

Since the selection of a clustering algorithm depends on the specific problem to be addressed, the aircraft seat system structure is examined. The desirable properties of a clustering algorithm include scalability, its ability to handle different data types, noise and outliers, and its interpretability and usability. The seat system is considered as a multi-scale, dynamic network of interconnected entities, yet follow a hierarchical structure. Therefore, Pizzuti's graph partition approach is a good fit for the CIAF and could also apply to other types of large-scale hierarchical systems. The DSM of the seat system was input into Pizzuti's multi-objective genetic algorithm to identify Pareto front solutions of communities in the architecture. The solution is selected based on the highest Newman Girvan modularity score, Q from Equation 6: 0.5619. The solution is shown in Figure 14 where six communities are identified. Figure 15 provides a closer look at the performance measures associated to each community by hiding the DSM cells in the spreadsheet.

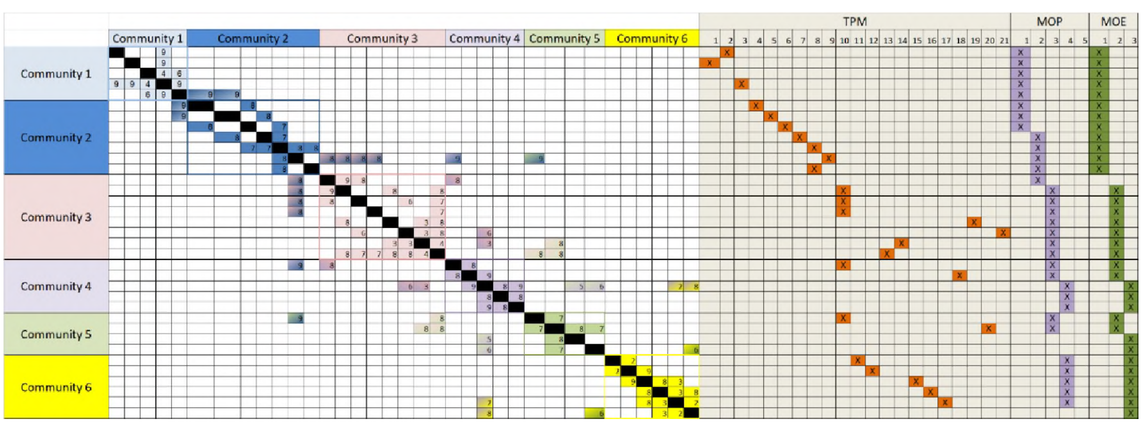

Figure 14. Clustered seat system and performance measures.

Community	TPM																					MOP					MOE		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	1	2	3	4	5	1	2	3
1	X	X	X																			X					X		
2				X	X	X	X	X	X													X	X				X		
3										X			X	X					X		X			X				X	
4										X								X						X	X			X	X
5										X										X				X				X	X
6											X	X			X	X	X								X				X

Figure 15. Identification of performance measures associated to communities.

Figure 16 provides a closer view of what performance measures are associated to community 6. For community 6, there are six components – Z, AA, AB, AC, AD, AE – each containing TRL values 6, 5, 7, 5, 3, & 9, respectively. The binary numbers in the DSM indicate an interface between system elements. This was used as an input to the clustering algorithm where the binary values are replaced with IRL values to calculate the CML. Blank boxes indicate there is no interface and thus, no IRL input is provided. TPMs 11, 12, 15, 16, and 17 are identified in this community and are associated to components Z, AA, AB, AC, and AD respectively whereas AE is associated to MOE 3 directly. In this case, the system element AE is critical at the customer-level, is qualitatively measured, and does not have a TPM nor MOP associated. The next step is to dive into these communities and verify the interfaces and assess maturity and risks prior to integration with other components within the community and with other communities.

				TRL						TPM					MP	ME
Community				6	5	7	5	3	9	11	12	15	16	17	4	3
6	Component	Z	6		2					0.2					0.2	0
		AA	5	2		5					0.5				0.2	0
		AB	7		5		5	3				0.75			0.2	0
		AC	5			5		3	5				0.45		0.2	0
		AD	3			3	3		2					0.4	0.1	0
		AE	9				5	2							0.2	0

Figure 16. Community 6 Values.

4.4. STEP 4: SCRUTINIZE COMMUNITIES AND ASSESS INTERFACES

Each of the six communities are placed in a tabular form as shown in the Figure 16 snapshot. Also, the min TRL-min IRL model is constructed for each community. Table 3 presents the min TRL-min IRL model for Community 6.

Table 3. Min TRL-Min IRL model for Community 6.

Level	1	2	3	4	5	6	7	8	9
TRL	0	0	1	0	1	1	1	0	1
IRL	0	3	3	1	5	0	0	0	0

The community maturity level for community 6, $CML_6 = 0.20$. By contrast, the calculated SRL is 0.38. Some components within a community interface with other communities, such as the airbag system making contact with the passenger. Table 3 presents the min TRL-min IRL model for intra-community interfaces between Community 4 where the airbag system resides and Community 6 where the passenger as an object in the seat system resides. The intra-community maturity level is 0.24, in contrast to the SRL of 0.52, which indicates there is an interface risk that a system is not ready for technology development and integration.

In the seat development plan, developmental tests were scheduled three months after CDR (Seat Supplier, 2013), which was a concern among the seat subject matter experts consisting of a technical program manager, an electrical engineer, a payloads engineer, and a certification engineer. They determined that even though CDR is “pencils down" on the design, the specification of the airbag component and its method of deployment had not been determined, the official Federal Aviation Administration (FAA) policy/guidance for making an assessment was not provided, and there were no data on the current design from developmental test. Because of this, they could not assess and conclude compatibility between the passenger and the airbag system during 16g crash scenarios. Therefore, the CML value of 0.24 is more representative of the readiness than the SRL value of 0.52 which indicates the end of technology development.

The original concept was to use the airbag to protect the passenger from injury in a 16g crash scenario, but past experiences and data indicate that airbag designs produce unpredictable results in protecting the head, lumbar, and neck. The experts' position at CDR led to discussing whether to 1) move the CDR out to after developmental testing with current airbag specifications and build a recovery plan to meet production schedule, 2) change the angle of the seat and/or pitch to be more similar to previously certified designs, 3) change the restraint system (e.g. 3-point harness) to mitigate risk of introducing unpredictability of airbag designs in protecting the occupant, 4) add other energy-absorbing materials on the impact interface, or 5) a combination of all of the above (Seat Supplier, 2013, 2014). Ultimately, the seat designer assumed the risk of proceeding to inflatable restraint seat system developmental tests and added buffers in the schedule for recovery needs.

To validate the concern with this low community maturity score, the associated performance measures - TPMs and MOPs - are cross examined to determine the readiness of components and subsystems are in meeting the critical requirements associated to passenger survivability. Since developmental tests were not performed by

Critical Design Review, there was no evidence that the current airbag specifications and seat design protect the 16g test dummy from injury. This means that the CML score of 0.24 is a more reliable metric than the SRL score of 0.52.

Table 4. Component interfaces between Communities 4 & 6.

Level	1	2	3	4	5	6	7	8	9
TRL	0	0	1	0	0	0	0	0	2
IRL	0	1	0	0	0	1	0	1	0

5. DISCUSSION

Assessing integration risks of an aircraft with 10^5 interfaces that is controlled by various software artifacts during development is extremely difficult. The CIAF case study of a seat system demonstrates that the clustering of interfaces into communities allows reasonable focus and assessment on integration risks based on the analysis of the readiness of the interfaces and defined performance measures within and across communities. The results demonstrated that the passenger control unit (PCU) software in the seat system was clustered in Community 3 with the electronic and power regulating elements that is controlled by the PCU interface. The seat system example was representative of an actual system design where the experts' position during critical design review was to refine the concept's approach to certification before moving into technology development and integration. The CIAF has demonstrated through CML and performance measure analyses that the experts' position were correct that the system concept was not ready for development and integration.

5.1. VALIDATION OF RESULTS AND METHODOLOGY

The decomposition of the seat system attributes into technical performance measures was done by seat functional experts and measures of performance of the system at different levels. The decomposition of the seat MOPs, such as passenger survivability, into a TPM, such as head injury criteria were validated against published regulatory requirements and by subject matter expert input. However, the performance values particularly measure of performance were based on judgement as to how far the requirement they believe will be met and thus, these values require more rigorous validation. There is a possibility that the level of effort to achieve 100% performance could take months of development and testing. It is suggested that the CIAF provides a defensible rationale for reevaluating ill-defined interfaces by virtue of the community's inability to adequately fulfill a contributing technical performance measure such as protecting the passenger in survivability tests.

The ability of the clustering algorithm to break down the network into communities depends on the validity of the objectives of the algorithm, in this case the fitness function. How the network is decomposed to solve a problem depends on how the problem is defined. The aircraft seat system was viewed as a dendrogram and thus, Pizzuti's hierarchical clustering algorithm was selected. Subject matter expert input is used to validate that the selected solution based on modularity score is a good representation of real collaborations between seat system interfaces. These subject matter experts were identified to have experience in payload and electrical design and integration at seat and airplane levels. The key question asked was:

- Is the selected community with the highest modularity score defensible as "valid" community to assess the interfaces and performance measures? ie., is it representative of real collaborations between corresponding airplane and seat interfaces?

Finally, from a risk perspective, the results of the CML assessment on a seat system using inputs from the critical design review (CDR) were compared with the SRL results. The experts (a technical program manager, payloads and electrical engineers, and a certification engineer) determined that at the CDR, the airbag system detail and interface definition as well as the overall restraint system and interface with the seat system were not sufficient to proceed into the detail design and testing phases due to significant risk of failure driving expensive, long-lead rework and tests.

5.2. CONCLUSION

The CIAF can be utilized at all stages during the system lifecycle with DSM input as it evolves, especially prior to component and subsystem integration and systems verification. System architects usually produce a SV-3 matrix (DoD, 2010) that summarizes system interactions. Even though detailed-level interfaces in this phase are not usually available (Dauby & Dagli, 2009), this SV-3 matrix deliverable can still be input into the CIAF to assess system-level interfaces and associated performance measures. Interface assessments influence the technical baseline, albeit fuzzy, as the system design evolves with more detail. The CIAF can be continuously used as the system design matures throughout the systems “V” process before the integration stages.

As the system matures, the DSM matures with additional components and interfaces included. When large-scale systems like an aircraft go into the detailed design phase, the DSM becomes difficult to comprehend and assess due to its complexity and scale. It is recommended to maintain a single source database to feed multiple DSMs from each subsystem element. The CIAF allows systems engineers to verify whether they missed any interfaces and identify any potential performance issues. The CML measure, when cross checked with the identification and quantification of key interface performance measures, provides better information on system readiness before systems integration than using SRL values.

5.3. SUMMARY

The CIAF supplements systems engineering with a methodology that facilitates the assessment of integration risks in large-scale and complex systems. The CIAF is used to determine if we have included the interfaces and if these interfaces have matured to a level of acceptable risk at a point in the development lifecycle. The first step in the process of using the CIAF is to identify the system of interest and generate a design structure matrix (DSM) that feeds into the multi-objective genetic algorithm (MOGA-net). MOGA-net uses equations 2 through 5 to divide the DSM (system of interest) into groups of nodes with dense, internal connections and uses the modularity score (equation 6) to select a solution out of several possible community structures. The CIAF in this study was used on a power seat system in a commercial aircraft containing hardware and software components such as power distributors and converters, actuators, airbag system, in-flight entertainment, and passenger control units. The community maturity level (equation 7) was used to determine the technical maturity of each community structure using the Min TRL-Min IRL model approach (Figure 6). Though the CML equation normalizes the resulting values that allows us to use a risk matrix approach, fine tuning of the equation is needed and is a future work consideration. Finally, the CML analysis is cross-examined with the analysis of technical performance measures (equation 9) to validate where the community of interface components with a level of maturity stand in meeting performance targets.

One of the primary objectives of the framework is to mitigate integration risks. To put together a plan to mitigate risks, one must understand the likelihood and consequences of an event if it were to occur and determine whether to reduce the likelihood of that event, reduce the severity of the consequence, or both. In the case of system integration risks, one of the ways to mitigate these risks is to identify and define performance measures for a key interface and verify the interface before systems integration. The framework, in conjunction with other systems engineering tools and methodologies, is an actionable approach to mitigating risks by cross-examining community maturity levels of system communities and associated performance measures. Furthermore, Model-Based Systems Engineers are encouraged to use the CIAF to evaluate interfaces in logical, functional, and physical models and feedback the analysis to these models.

6. FUTURE WORK

While the research focused on the seat system as a case study, it is envisioned that the framework could be applicable at different scales and complexity of systems in other technical domains such as residential and commercial power systems for a region or a large-scale software system. There is opportunity to evaluate the applicability of the CIAF on other system problems of varying scale and complexity.

Community detection in complex networks has gained significant attention and while the MOGA-Net algorithm is effective in detecting real-world communities, there is an opportunity to explore other and more effective algorithms for the CIAF to use to address specific large-scale system problems. Furthermore, while scoring integration readiness is based on the verification of interface requirements, the SME's judgment and assessment on IRL level (and performance measures) may differ. Use of a Bayesian network and probability distributions may provide consistent and mathematically rigorous validation of the confidence level among experts on the IRL level, allowing a better perspective on the system integration risks. Cardinal coefficients for TRLs (Fahimian & Behdinan, 2017; Revfi, Wilwer, Behdinan, & Albers, 2020) based on the Analytic Hierarchy Process (AHP) have been used to characterize technology readiness level coefficients for design, which may improve the quality and accuracy of the CML metric, performance measures, and risk analysis. Perhaps when using judgement on performance measures, a fuzzy inference system (FIS) with a set of rules can capture inputs on MOPs and MOEs and convert them into crisp values.

The inclusion of this framework into a Model-based systems engineering (MBSE) method and/or tool would have interesting implications. Linking the DSM information to SysML diagrams or OPM model would allow changes to the system design to be automatically updated in the DSM, thus modifying the inputs to the CIAF and possibly changing the overall readiness level. Since the value of MBSE depends on the quality of the model including the information in the DSM, there is future work to create an automated feedback loop from the interface analysis of the DSM in the CIAF to the SysML models.

REFERENCES

Arena, M. V, Younossi, O., Brancato, K., Blickstein, I., & Grammich, C. A. (2008). Why has the cost of fixed-wing aircraft risen? A Macroscopic Examination of the Trends in U.S. Military Aircraft Costs over the Past Several Decades. In *NATIONAL DEFENSE RESEARCH INSTITUTE and PROJECT AIR FORCE*.

Bilbro, A. (2007). *Systematic Assessment of the Program/Project Impacts of Technological Advancement and Insertion*. Huntsville, AL.

Burkhart, R., Friedenthal, S., Griego, R., Sampson, M., & Spiby, P. (2007). Model Based Systems Engineering (MBSE) Track. *INCOSE International Symposium*. https://doi.org/10.1002/j.2334-5837.2007.tb02999.x

Carlson, J. M., & Doyle, J. (2002). Complexity and robustness. *Proceedings of the National Academy of Sciences*. https://doi.org/10.1073/pnas.012582499

Carson, R. S. (2015). MBSE Implementation across Diverse Domains. *INSIGHT*. https://doi.org/10.1002/inst.12016

Conrow, E. H. (2011). Estimating Technology Readiness Level Coefficients. *Journal of Spacecraft and Rockets*, *48*(1), 146–152. https://doi.org/10.2514/1.46753

Corns, S., & Gibson, C. (2012). A model-based reference architecture for medical device development. *22nd Annual International Symposium of the International Council on Systems Engineering, INCOSE 2012 and the 8th Biennial European Systems Engineering Conference 2012, EuSEC 2012*. https://doi.org/10.1002/j.2334-5837.2012.tb01457.x

Dabkowski, M., Valerdi, R., & Farr, J. (2014). Exploiting architectural communities in early life cycle cost estimation. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2014.03.013

Dauby, J. P., & Dagli, C. H. (2009). Using extensible modeling in systems engineering and architectural search. *2009 IEEE International Systems Conference Proceedings*. https://doi.org/10.1109/SYSTEMS.2009.4815840

Department of Defense. (2008). Systems Engineering Guide for Systems of Systems. In *Technology*. https://doi.org/10.1109/EMR.2008.4778760

DoD. (2010). DoDAF Architecture Framework Version 2.02. *Department of Defense, USA*.

Eppinger, S. D., & Browning, T. R. (2012). Design structure matrix methods and applications. In *Engineering systems*. https://doi.org/10.1037/a0029756

Estefan, J. A. (2007). Survey of Model-Based Systems Engineering (MBSE) Methodologies. *Incose MBSE Focus Group*. https://doi.org/10.1109/35.295942

Fahimian, M., & Behdinan, K. (2017). On characterization of technology readiness level coefficients for design. *Proceedings of the International Conference on Engineering Design, ICED*.

Friedenthal, S., Moore, A., & Steiner, R. (2014). A practical guide to SysML: The systems modeling language. In *Systems Engineering*. https://doi.org/10.1016/B978-0-12-800202-5.00002-3

GAO. (2015). Assessments of Selected Weapon Programs. *Report to Congressional Committees*.

GAO. (2018). *James Webb Space Telescope: Integration and Test Challenges Have Delayed Launch and Threaten to Push Costs Over Cap*.

Kapurch, S. J. (2010). *NASA Systems Engineering Handbook*.

Kujawski, E. (2013). Analysis and critique of the system readiness level. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*. https://doi.org/10.1109/TSMCA.2012.2209868

Lancichinetti, A., & Fortunato, S. (2009). Community detection algorithms: A comparative analysis. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*. https://doi.org/10.1103/PhysRevE.80.056117

Long, D., & Zane, S. (2011). *A primer for model-based systems engineering*. Blacksburg, VA.

Madni, A. M., & Sievers, M. (2018). Model-based systems engineering: Motivation, current status, and research opportunities. *Systems Engineering*. https://doi.org/10.1002/sys.21438

Magnaye, R. B., Sauser, B. J., & Ramirez-Marquez, J. E. (2010). System development planning using readiness levels in a cost of development minimization model. *Systems Engineering*. https://doi.org/10.1002/sys.20151

Malone, R., Friedland, B., Herrold, J., & Fogarty, D. (2016). Insights from Large Scale Model Based Systems Engineering at Boeing. *INCOSE International Symposium*. https://doi.org/10.1002/j.2334-5837.2016.00177.x

McConkie, E., Mazzuchi, T. A., Sarkani, S., & Marchette, D. (2013). Mathematical properties of System Readiness Levels. *Systems Engineering*. https://doi.org/10.1002/sys.21237

McDaniel, R. R., Lanham, H. J., & Anderson, R. A. (2009). Implications of complex adaptive systems theory for the design of research on health care organizations. *Health Care Management Review*. https://doi.org/10.1097/HMR.0b013e31819c8b38

Mitchell, M. (1998). An Introduction to Genetic Algorithms (Complex Adaptive Systems). *The MIT Press*.

NASA. (2016). *Expanded Guidance for NASA Systems Engineering, Volume 1: Systems Engineering Practices*. NASA.

Newman, M. E. J. (2004). Detecting community structure in networks. *The European Physical Journal B - Condensed Matter*, *38*(2), 321–330. https://doi.org/10.1140/epjb/e2004-00124-y

Oliver, D. W., Kelliher, T. P., & Keegan, J. G. (2004). Engineering Complex Systems with Models and Objects. *New York*.

Pizzuti, C. (2012). A multiobjective genetic algorithm to find communities in complex networks. *IEEE Transactions on Evolutionary Computation*. https://doi.org/10.1109/TEVC.2011.2161090

Revfi, S., Wilwer, J., Behdinan, K., & Albers, A. (2020). DESIGN READINESS OF MULTI-MATERIAL CONCEPTS: MANUFACTURING AND JOINING TECHNOLOGY INTEGRATED EVALUATION OF CONCEPT MATURITY LEVELS USING CARDINAL COEFFICIENTS. *Proceedings of the Design Society: DESIGN Conference*. https://doi.org/10.1017/dsd.2020.274

Sauser, B., Gove, R., Forbes, E., & Ramirez-Marquez, J. E. (2010). Integration maturity metrics: Development of an integration readiness level. *Information Knowledge Systems Management*. https://doi.org/10.3233/iks-2010-0133

Sauser, B. J., Marquez, J. E. R., Henry, D., & DiMarzio, D. (2008). A system maturity index for the systems engineering life cycle. *International Journal of Industrial and Systems Engineering*. https://doi.org/10.1504/IJISE.2008.020680

Seat Supplier. (2013). *Personal Communication*.

Seat Supplier. (2014). *Personal Communication*.

Tamaskar, S., Neema, K., & Delaurentis, D. (2014). Framework for measuring complexity of aerospace systems. *Research in Engineering Design*. https://doi.org/10.1007/s00163-014-0169-5

Voirin, J.-L., Bonnet, S., Normand, V., & Exertier, D. (2015). From initial investigations up to large-scale rollout of an MBSE method and its supporting workbench: the Thales experience. *INCOSE International Symposium*. https://doi.org/10.1002/j.2334-5837.2015.00066.x

Yasseri, S., & Bahai, H. (2018). System Readiness Level Estimation of Oil and Gas Production Systems. *International Journal of Coastal and Offshore Engineering*. https://doi.org/10.29252/ijcoe.2.2.31

Yasseri, S. F., & Bahai, H. (2020). Case studies in estimating subsea systems' readiness level. *Underwater Technology*. https://doi.org/10.3723/ut.37.013

III. FUZZY-RISK ASSESSMENT METHODOLOGY FOR LARGE-SCALE SYSTEMS

Josh Goldschmid* and Steven M. Corns

Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, USA

*jhgrm5@umsystem.edu

Deparment of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409

ABSTRACT

Integration readiness analysis, often neglected in aircraft design, is a difficult systems engineering task but critical in the mitigation of integration risks in large-scale, complex systems. This paper offers engineering managers a soft computing approach that compliments risk management standards to measure integration readiness risks and to appreciate the nuances of integration risks within a set of highly interconnected elements in a system. The approach uses community detection algorithms to explore the population of large-scale system elements and aggregate densely interfacing entities into communities and then uses a fuzzy inference system to evaluate the integration risks of each community. The resulting risk values of each community are placed on a 5x5 risk matrix for engjneering management reviews and decision-making.

1. INTRODUCTION

1.1. BACKGROUND

Integration readiness analysis is a difficult systems engineering task that if not conducted sufficiently early in a large-scale system development lifecycle, systems integration that occurs late in the development process is at considerable risk of failure with severe cost and schedule consequences (GAO, 2015, 2018). These consequences caused by inadequate interface definition is usually manifested during the systems verification phase. This occurs often in the aerospace sector, where the threats are a combination of quickly evolving customer requirements and technologies, variety of customer expectations, and rigorous time frame for product development that can have a significant impact on risk level (Jaifer et al., 2020).

There is an inherent bias to underappreciate external threats and challenges that potentially disrupt schedule and cost plans that is common among the development of complex systems (Reeves et al., 2013). Though the external threats in this context can mean accounting changes, supplier changes, program requirement changes, economic and political issues, or other forces of nature, the authors believe there is inherent bias to underappreciate integration threats of technical interfaces outside of a typical product breakdown structure that a design team may have overlooked. The external technical interfaces are for example, the electrical connections needed for fire detection in an aircraft lavatory. These external interfaces are often neglected in aircraft design (Jackson, 2016).

This paper offers engineering managers a soft computing approach that compliments risk management standards (particularly PMI and ISO) to measure integration readiness risks and to appreciate the nuances of integration risks within a set of highly interconnected elements in a system. The approach uses community detection algorithms to explore the population of large-scale system elements and aggregate densely interfacing entities into communities and then uses a fuzzy inference system to evaluate the integration risks of each community. The resulting risk values of each community are placed on a 5x5 risk matrix for engjneering management reviews and decision-making. The approach was tested on a commercial aircraft seat system with data from the Critical Design Review.

1.2. REVIEW OF LITERATURE

1.2.1. Risk Assessment. The traditional measure of risk is the product of probability, P, and the weight of an adverse consequence, C (Crouch, E A.C.;Wilson, 1982). In many industries where project management practices are utilized, the Risk Score formula is used (Project Management Institute (2018):

$$Risk = P \times C \tag{1}$$

One of the common and popular approaches used to characterize and prioritize risk is the risk matrix (Figure 1) which uses crisp inputs from Equation 1. Caution is needed when using this formula because when multiplying probability and consequence values with ordinal scales, it may produce a significant figure that has little value to a project manager or a chief engineer. For instance, the formula will indicate that events of low probability with large consequences have the same considerations as an event likely to happen but with lesser consequences. Furthermore, multiplying ordinal numbers is mathematically invalid such that a consequence that is assigned a 4 on an ordinal scale is not necessarily twice as consequential as a consequence with a score of 2 (Hubbard & Evans, 2010). However, this does not mean risk scores are not useful, but avoid using these scores in isolation to make decisions (Kaplan & Garrick, 1981).

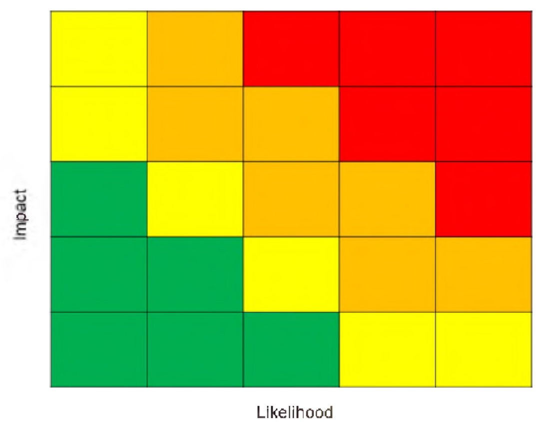

Figure 1. Standard Risk Matrix

1.2.2. Department of Defense Risk Assessment. The DoD Risk, Issue, and Opportunity (RIO) guide suggests a comprehensive set of approaches to inquire, examine, and analyze risks such as interviewing technical experts, identifying dependencies and interoperability requirements, assessing maturity of critical technologies, analyzing metric trends such as technical performance measures (TPMs), and performing non-advocate reviews (NIST, 2012; Office of the Deputy Assistant Secretary of Defense for Systems Engineering, 2017). These risks are usually ranked and prioritized before the risk handling process where actions are in place to address these risks.

The DoD and many private firms report the risks in a 5x5 risk cube with color coded areas (Figure 2) that represent the rating or prioritization of an identified risk. The risk matrix maps the impact (consequence) and likelihood into a common space. For example, the upper right corner shows high likelihood (5) and high impact (5) with crisp (non-fuzzy) risk score of 25 (Equation 1).

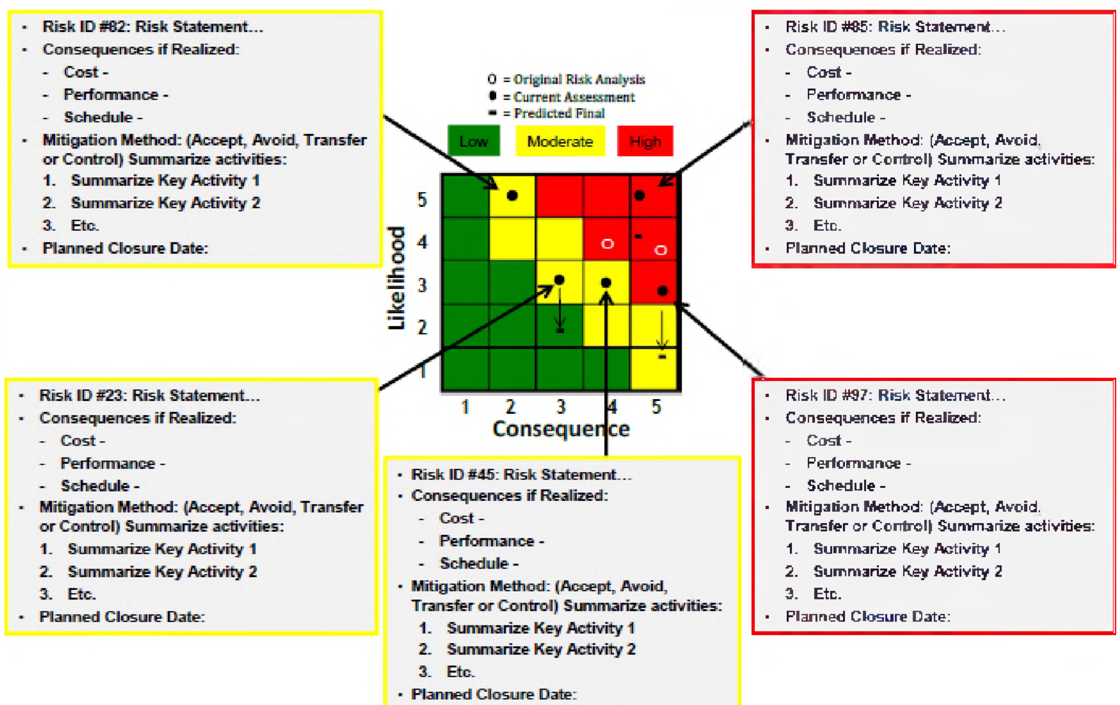

Figure 2. DoD Suggested Risk Reporting Format (Office of the Deputy Assistant Secretary of Defense for Systems Engineering, 2017)

1.2.3. Technology Readiness and Risk Assessment. Mankins (2009) reformulated the traditional risk matrix with a quantitative Technology Readiness and

Risk Assessment (TRRA) model that includes technology readiness level (TRL), the degree of difficulty of moving technology from one TRL to another, and Technology Need Value (TNV). The model incorporates these values into a technology risk matrix with probability of failure on the y-axis and consequence of failure on the x-axis. The TRL scales, originally developed by National Aeronautics and Space Administration (NASA), range from 1 to 9 indicating increasing maturity and technology risks. Bilbro (2007) proposed the technology maturity assessment (TMA) is an assessment technique that uses a work breakdown structure (WBS) to identify key technologies subject to the TRL maturity scale. Bilbro argues that the maturity of a TRL 9 technology drops to TRL 5 when it is integrated into a new environment or configuration. For most large and complex systems there are too many WBS elements to address and track individually and so a metric is needed to capture readiness of these systems.

To address the limitations of TRLs, Sauser (2008) proposed the System Readiness Level (SRL) metric to provide a system-level view in real-time of the system development and maturity in relation to the Department of Defense's (DoD) Phases of Development, giving managers opportunities to take proactive measures to reduce developmental risk. The SRL metric introduced Integration Readiness Level (IRL) variables to determine the overall SRL. The IRL is a measurement of the interface compatibility indicating maturity between interfacing elements (B. Sauser et al., 2010). IRLs scale similarly to TRLs but start with zero instead of one, with zero indicating there is no interface or no integration has been planned or intended. Figure 3 shows an example of a system under consideration (real system examples found in Yasseri & Bahai [2018] and Yasseri & Bahai [2020]), where the TRL for each of the subsystems is identified as

A, B, C, and D. The IRL represents the maturity of the interfaces between these subsystems. These subsystems are represented in equation 2 with the normalized IRL matrix capturing the interface maturity and the TRL capturing the subsystem maturity where the overall SRL is 0.49, indicating system maturity in the Technology Development phase (Magnaye et al., 2010).

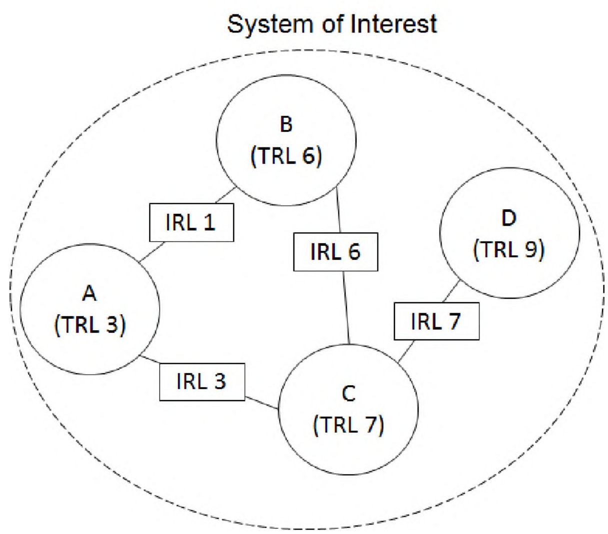

Figure 3. System of Interest Readiness Level

$$SRL = \overset{\text{Normalized IRL Matrix}}{\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}\begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 1 & 1/9 & 3/9 & 0 \\ 1/9 & 1 & 6/9 & 0 \\ 3/9 & 6/9 & 1 & 7/9 \\ 0 & 0 & 7/9 & 1 \end{bmatrix} \end{array}} \times \overset{\text{TRL Matrix}}{\begin{pmatrix} 3/9 \\ 6/9 \\ 7/9 \\ 9/9 \end{pmatrix}} = 0.49 \qquad (2)$$

The ordinal scales of TRLs and IRLs do not convey information about the degree of differences between measures, making arithmetic calculations with these scales of limited utility (Conrow, 2011b; Kujawski, 2013; McConkie et al., 2013).

1.2.4. Other Risk Estimation Techniques. Garg (2017) proposed an objective risk estimation technique using the product of TRL values as a measure for likelihood and a network connectivity metric to estimate impact on the system architecture. It is intuitive to gage change impact on the system architecture based on propagation of interfaces of components across the system. Therefore, a network connectivity metric is used to measure impact propagation throughout the architecture. The authors proposed that since TRLs are good estimators of uncertainty in technology readiness, the inversion of TRL scales 1-9 are used as the basis for likelihood scores. To provide information about the interfaces that each component has, the authors combined the risk score information with a Design Structure Matrix (DSM) view of the system. The risk score for the interfaces is then calculated using the formula:

$$Interface\ Risk_{ij} = \max(L_i, L_j) * \max(I_i, I_j) \tag{3}$$

Where L_i and L_j represent the likelihood scores for the interfacing components and I_i and I_j as impact scores for each component. As implied earlier, arithmetic calculations with ordinal scales have limited utility and it is difficult to understand the implications of a component with a specific maturity has on the overall integration risk.

Clarkson (2004) developed a method to predict change propagation in a complex design and obtained a risk matrix for the system. In this method, practitioners, using experience, judgment, and documentation, performed four assessments for each pair of interfacing components: the likelihood that a component change will propagate through their interfaces and the impact of the change propagation. A change propagation tree is generated to sum up values of likelihood and impact scores. This effort is extremely intensive for large-scale systems with complex networks.

1.2.5. Summary. This research reviews the challenges in assessing integration risks in large-scale and complicated systems using concepts of technical maturity, system architecture, and interface analysis while keeping the assessment effort practical for application. It is incumbent on the system architect to continuously monitor the risks of the system during development to ensure compliance with key system-level requirements and performance measures.

2. APPROACH

NIST (2012) noted that:

> *Risk assessments are often not precise instruments of measurement and reflect: (i) the limitations of the specific assessment methodologies, tools, and techniques employed; (ii) the subjectivity, quality, and trustworthiness of the data used; (iii) the interpretation of assessment results; and (iv) the skills and expertise of those individuals or groups conducting the assessments.*

In the field of soft computing, intelligent systems techniques have proven to be effective in addressing a range of complex problems dominated by uncertainty and available, imprecise information (Ibrahim, 2016; Konar, 1999). The methodology proposed in this paper recognizes that, since fuzzy logic (Zadeh, 1975) provides a framework for approximate reasoning where information is subjective, incomplete, or uncertain, it has a potential role in the integration risk assessment of large-scale, complex systems where the probability assessment is based on expert opinion and where the risk space is multidimensional and nonlinear (Marchetti, 2012).

The fuzzy risk assessment methodology (FRAM) proposed in this paper asserts that the system integration risk is a characterization of:

- Technical maturity of clustered components for integration
- Performance of aggregated components and;
- Current system development phase

The FRAM calculates the integration risks with two inputs from the clustering-based interface assessment framework (CIAF) (FRAMEWORK DESCRIPTION) shown in Figure 4: community maturity level and performance measures. The CIAF is a two-stage process resulting in a thorough integration readiness analysis of complex interfaces in a system. The first stage establishes the communities using Pizzuti's evolutionary-based clustering algorithm (MOGA-Net) (Pizzuti, 2008) to identify a solution set of communities from the Design Structure Matrix (DSM) of interfacing components and a solution is selected using Newman-Girvan's modularity metric (Newman, 2004b). In the second stage of the framework, each of the communities from the selected solution are assessed for integration readiness using the Community Maturity Level (CML) metric. The community maturity is then cross-examined with the community's performance measures as another layer of analysis to validate integration readiness.

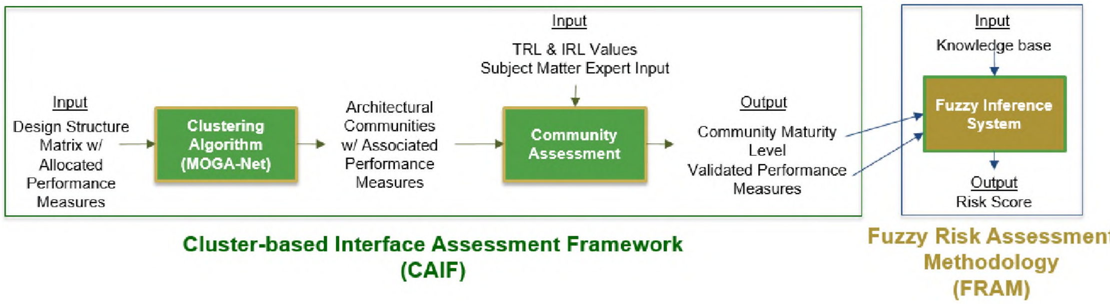

Figure 4. Cluster-based Interface Assessment Framework

The crisp inputs of community maturity levels and performance measures are fed into the Mamdani fuzzy inference system (FIS) with a set of rules to calculate the overall integration readiness risk score based on current developmental phase, maturity, and performance. The risk score for each community is placed on a 5x5 risk matrix (Figure 5) which is elaborated in below sections.

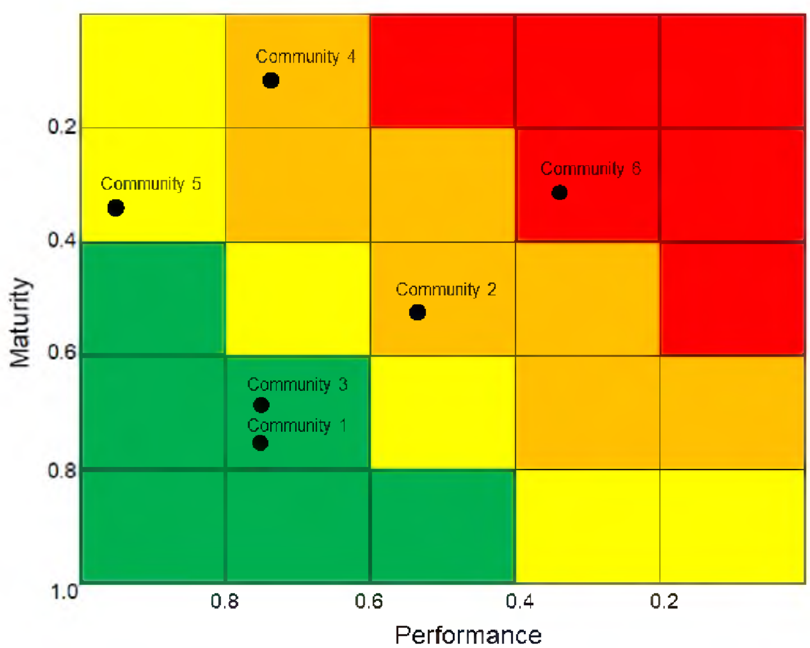

Figure 5. 5x5 Risk Chart and Community Integration Risk Scores during Critical Design Review (CDR).

2.1.1. Fuzzy Sets. There are two fundamental concepts of Fuzzy Set Theory (Zadeh, 1965), linguistic variables and fuzzy sets. The linguistic variables represent opinions that are usually comprehended by a typical audience. For example, the weather conditions can be described as "humid" or dry." Fuzzy sets are defined as a class of objects with a continuum of grades of membership between 0 and 1. To illustrate this in the context of the 5x5 risk matrix, the matrix bounds the risk level by considering the product between the likelihood of occurrence (1-5) and severity of consequence (1-5). Each risk product belongs to a specific category on the risk matrix as either "low," "moderate," "moderately high," and "critical." The fuzzy set is characterized by membership functions $\boldsymbol{\mu(x)}$ that assigns membership values between 0 and 1 to its components x:

$$\mu(x): X \rightarrow [0,1] \tag{4}$$

Applying the fuzzy set theory to the risk matrix results in a gradual and smooth transition between risk-level categories (Figure 6)

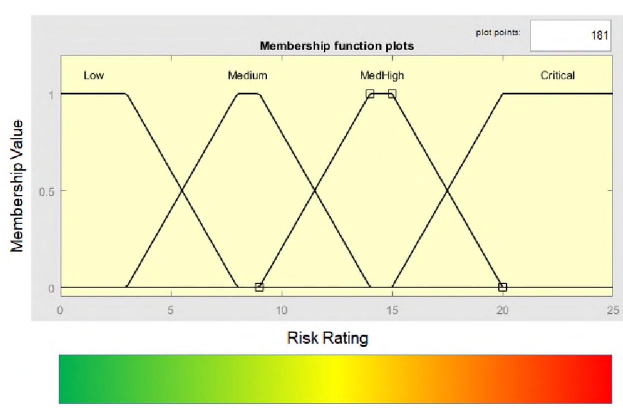

Figure 6. Graphical Representation of Fuzzy Risk Assessment Matrix

2.1.2. Fuzzy Characterization of Community Maturity Levels. DoD and private firm program management offices perform technical reviews in phases in the systems engineering lifecycle as a fundamental risk reduction process, adhering to standard requirements in ISO/IEC/IEEE 15288. The scale and definitions of the various levels of the CML are correlated to phases of the systems engineering lifecycle (Table 1) from Concept Refinement to Operations & Support. It is important to note that a community that has not reached full maturity is still capable of transitioning into the production phase at risk, with the caveat the key performance measures associated to the community demonstrates with a certain confidence level (acceptable risk) that the targets are on track to be met.

Table 1. Community Maturity Level Assessment with definitions and the associated phase of system development

CML	Phase	Definitions
0.90 to 1.00	**Operations & Support**	Execute a support program that meets material readiness and operational support performance requirements and sustains the community in the most cost-effectice manner over its total lifecycle
0.80 to 0.89	**Production**	Achieve operational capability that satisfies mission needs
0.50 to 0.79	**Engineering & Manufacturing Development**	Develop capability of clustered, interfacing technologies ; reduce integration and manufacturing risk; ensure operational supportability of the cluster; minimize logistics footprint; implement human systems integration; design for production; ensure affordability and protection of critical program informationl and demonstrate community integration, interoperability, safety, and utility
0.20 to 0.49	**Technology Development**	Reduce technology risks and determine appropriate set of technologies to integrate to serve key functions
0.10 to 0.19	**Concept Refinement**	Refine initial concept. Develop technology and interface strategy

2.1.3. Fuzzy Characterization of System Performance. Performance can be measured at different levels in the architecture including integrated elements such an aircraft cabin consuming power, but in a highly dense network with various key measures such as power consumption, safety and reliability, comprehensive analysis for risk assessment is crucial. Each community with dense interfaces identified in the system architecture may have an arrangement of performance measures, key performance attributes (KPAs), measures of effectiveness (MOEs), measures of performances (MOPs), and technical performance measures (TPMs) where the DoD engineering guidance suggests the decomposition of MOEs into MOPs that are subsequently supported by TPMs (NASA, 2016; Roedler & Jones, 2005).

Lesinski (2015) implied that a common challenge with the evaluation of an architecture is a comprehensive search of technical performance attributes across an exhaustive design space that are particularly fuzzy, especially in the early system development phases. Lesinski proposed a value focused thinking and fuzzy system approach to assess a system architecture that includes the customer's value input on the TPMs to convert them into a dimensionless scale from 0-100. The combined effects of the TPM to KPA tree characterizes the architecture's performance rating, using a set of fuzzy rules on KPA attributes. This paper utilizes a similar approach of the TPM conversion into linguistic variables using the FIS, where 0-49% is "Did not meet," 50-94% as "Somewhat Met," 95-99% as "Met," and anything above 100% as "Exceed." However, the crisp range to linguistic variables may depend on the customer value and acceptance range of the MOPs and MOEs that it traces to. This is an area of further research needed. In the FRAM, the fuzzy set rules characterize the integration risks of each cluster based on CML and performance measure levels in the cluster.

2.1.4. Overall Fuzzy Risk Assessment. The integration risk for each community is based on the fuzzy assessment of the community's maturity and performance at a point in time in the developmental lifecycle. MATLAB Fuzzy Toolbox (MathWorks 2019) is used to program the fuzzy assessment framework where the community maturity level and performance measures are inputs (Figure 7) with unique membership functions. Triangular and trapezoidal membership functions (MF) are adopted for the variables due to simplicity to implement and is computationally easy (Figures 8 to 10). The membership functions for maturity is asymmetrical to align with the maturity scores of the system development phases in Table 1 because typically more time and energy is required to architect and design a new product than to produce and support.

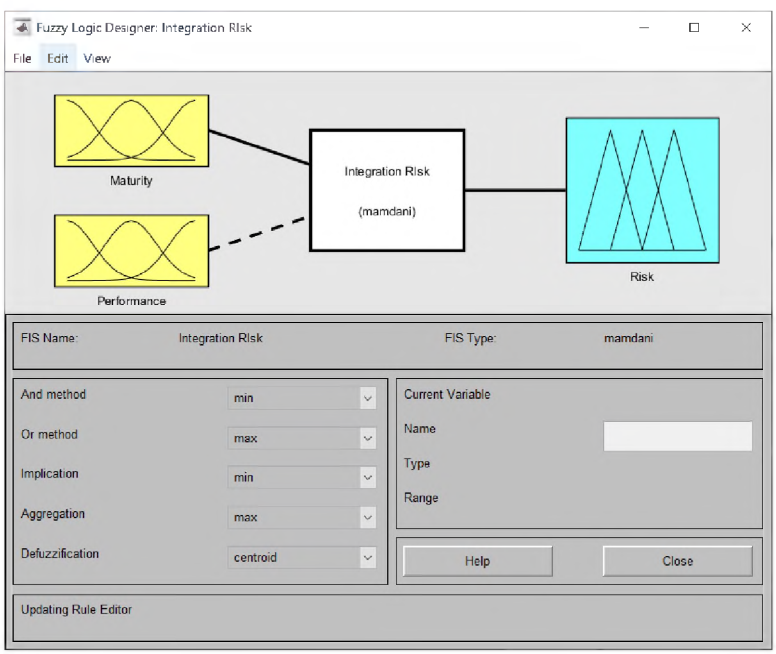

Figure 7. Fuzzy Inference System Parameters

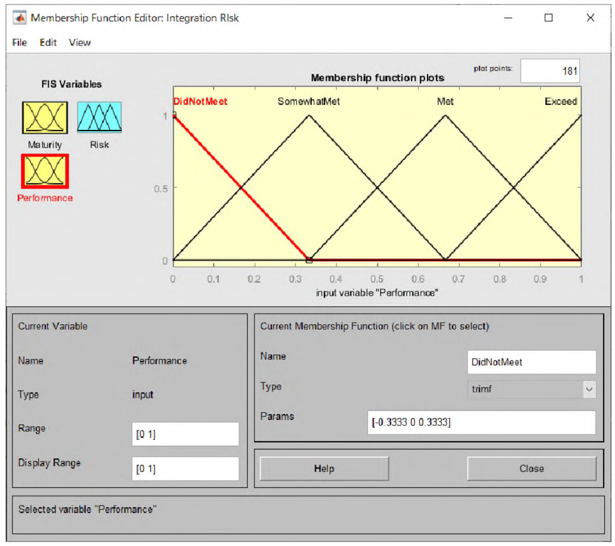

Figure 8. Membership Function for Performance

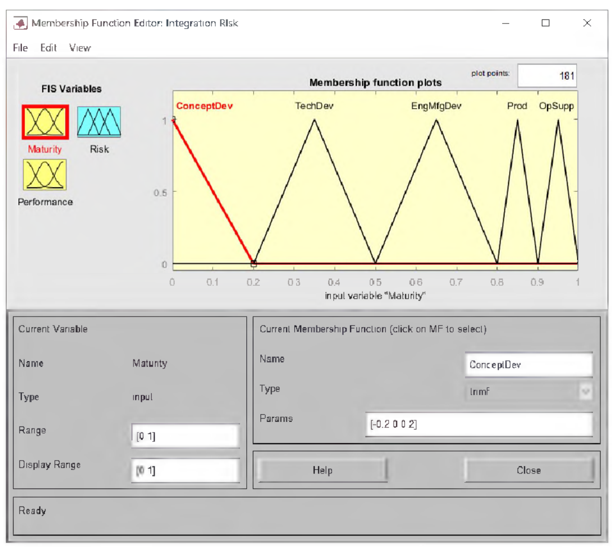

Figure 9. Membership Function for Maturity

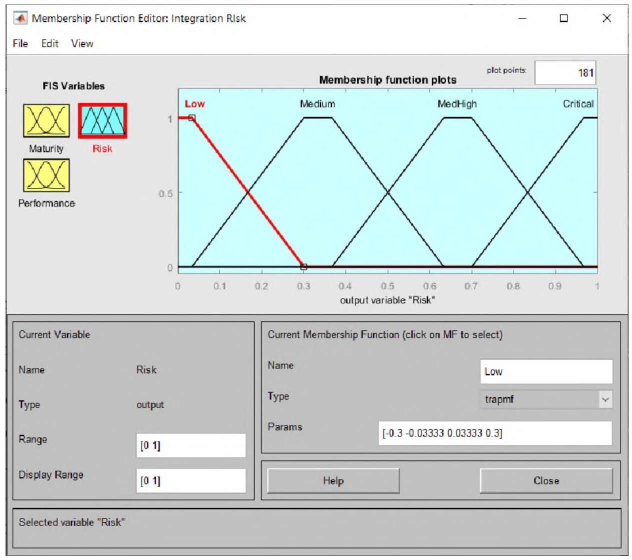

Figure 10. Membership Function for Risk Score

A set of fuzzy rules (Figure 11) are developed to characterize the overall integration risks of the system architecture based on the combined characterization of the CML and performance values.

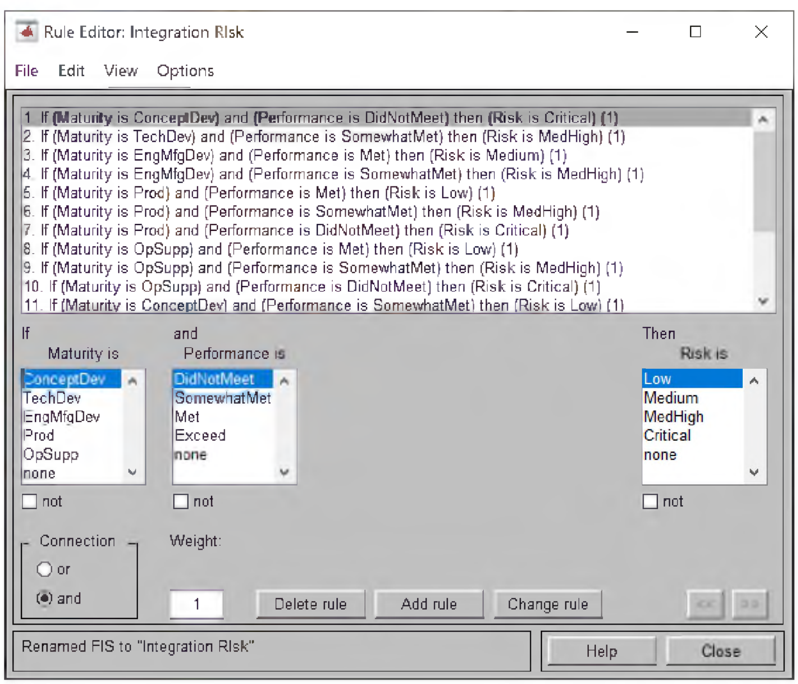

Figure 11. Fuzzy Rules

Figure 12 provides the surface output in 3D that consist of the two variables contributing to risk score. Performance is a clear indicator of the risks associated that for example, even if the community of densely interfaced components is mature but does not meet performance, there is a risk of critical rework to reach the level of required performance. For example, while the airbag system components were relatively mature at the Business Class Critical Design Review for an airline (Seat Supplier, 2014), the redesign and retest to pass the Federal Aviation Administration's (FAA) expectation for passenger safety took approximately four months to complete. If performance is high but maturity is low, it may take some steps to mature a product but high performance is an indicator of good confidence of the integration viability.

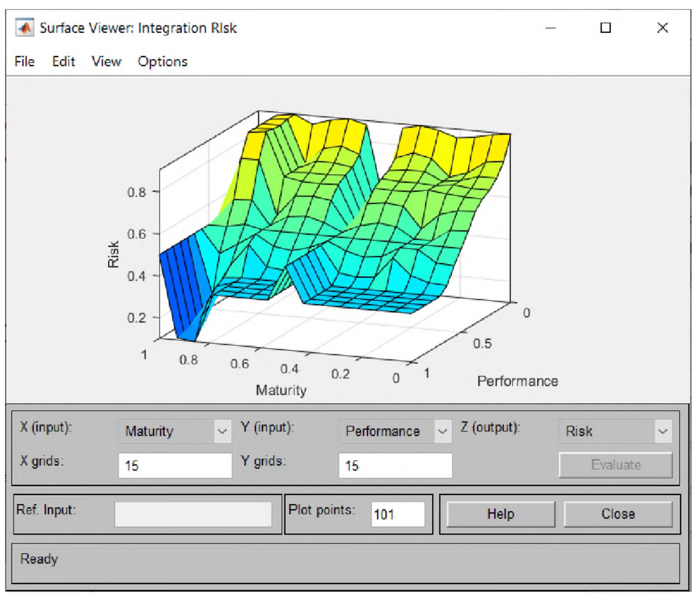

Figure 12. Surface Output of the Integration Risk Score

3. DISCUSSION

There are 20 conditions to satisfy the integration risk score output (Figure 13). In the rule viewer with an example where the overall maturity of the cluster is in the engineering development stage and performance is 0.182 (Met Some), the risk is considered Medium High (0.71). Before going into production, acceptable performance in the view of the customer should be demonstrated. As shown in Figure 5 for a seat system during the Critical Design Review, the Community 2 shows maturity of 0.55 (Engineering Development Phase) and Performance at 56% (Somewhat Met), which indicates Risk is 0.49 or Medium-High (Figure 10). There is perhaps time to mitigate integration risks before freezing the design, but if we were at Production Readiness Review (PRR), the risk of integration would be in the red domain on the risk chart.

The objective of this work is to provide soft-computing method using an intelligent system (Mamdani FIS) for analyzing integration risks over the developmental lifecycle. Assessing risks for a large-scale system can be tedious using traditional risk assessment techniques. The clustering technique using the CIAF breaks down the system into communities of dense interfaces which allows a more exhaustive analysis of the risks using the FRAM based on community maturity and performance levels in each community.

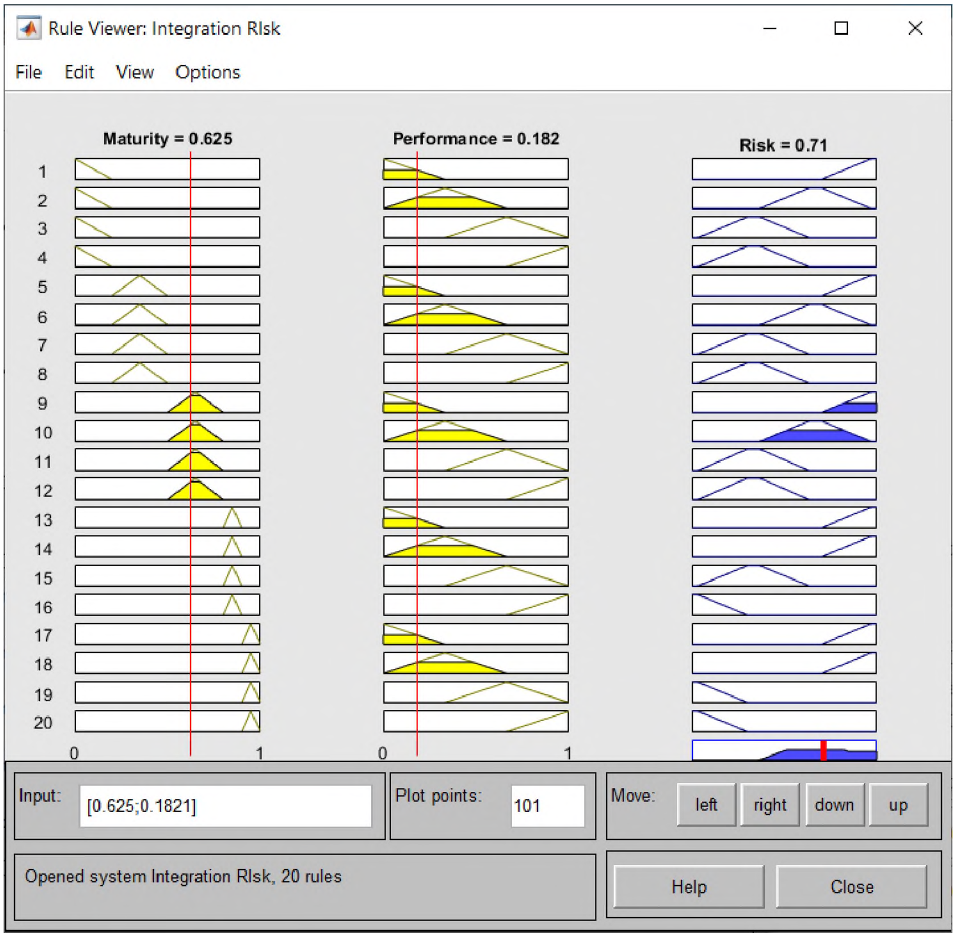

Figure 13. Risk Assessment Simulation - Rule Viewer

3.1. IMPLICATIONS FOR ENGINEERING MANAGERS

There is little research that rigorously validates the risk matrix performance in improving risk management decisions and there is a risk of using these risk matrices due to, for example, inconsistent risk score acceptance, centering bias, equating events with the same score, or whether it may be extremely arbitrary (Anthony Cox, 2008; Thomas, Bratvold, & Bickel, 2014). However, they are useful. In the F/A-18E/F Super Hornet program, the team that received system requirements via allocations used McDonnell Douglas's risk management process and analyzed them in terms of probability and consequence on the 5x5 risk matrix (Springsteen, Beth; Bailey, Elizabeth; Nash, Sarah; Woosley, 1999). The proactive, early identification of risks and weekly reporting using the 5x5 matrices were instrumental in the process, where it takes center stage at gate or technical reviews.

This study has a noteworthy implication for engineering managers. The use of tradional risk matrices based on probability of an event and consequence should an event to occur is exhausting for large-scale or complex systems where key interfaces that have consequences can be overlooked. The clustering techniques could help engineers focus their assessment on the highly interconnected elements that drive system performance and functional capabilities. This paper calls for engineers to have greater awareness of interconnected system elements through clustering and concludes with suggesting a soft computing approach technique for better assessment of interfaces that largely influence integration risks. The soft computing approach evaluates integration readiness of a set of communities in a large-scale system that drive key functions, where the the resulting integration risk values of each community based on maturity and performance are placed on a 5x5 risk matrix for engjneering management reviews and decision-making.

4. CONCLUSIONS AND FUTURE WORK

The reason for assessing integration risks of interfaces in a network is as Clarkson and Garg points out, change propagates between components through their interfaces. When estimating the impact of integration on the system architecture, it is reasonable to consider the architecture's network connectivity to improve our understanding of the integration risks and reduce bias of underappreciating external interfaces that should be evaluated with internal interfaces. The FRAM followed by the CIAF supplements systems engineering and engineering management with a methodology that facilitates the assessment of integration risks in large-scale systems that is important for system development milestone reviews.

While the research focused on the seat system as a case study, it is envisioned that the framework could be applicable at different scales and complexity of systems in other technical domains such as residential and commercial power systems for a region or a large-scale software system. There is opportunity to evaluate the applicability of the approach on other system problems of varying scale and complexity.

Furthermore, while scoring integration readiness is based on the verification of interface requirements, the SME's judgment and assessment on IRL level (and performance measures) may differ. Use of a Bayesian network and probability distributions may provide consistent and mathematically rigorous validation of the confidence level among experts on the IRL level, allowing a better perspective on the system integration risks. Cardinal coefficients for TRLs (Fahimian & Behdinan, 2017; Revfi et al., 2020) based on the Analytic Hierarchy Process (AHP) have been used to characterize technology readiness level coefficients for design, which may improve the quality and accuracy of the CML metric and risk analysis.

REFERENCES

Anthony Cox, L. (2008). What's wrong with risk matrices? Risk Analysis. https://doi.org/10.1111/j.1539-6924.2008.01030.x

Bilbro, A. (2007). Systematic Assessment of the Program/Project Impacts of Technological Advancement and Insertion. Huntsville, AL.

Clarkson, P. J., Simons, C., & Eckert, C. (2004). Predicting change propagation in complex design. Journal of Mechanical Design, Transactions of the ASME. https://doi.org/10.1115/1.1765117

Conrow, E. H. (2011). Estimating Technology Readiness Level Coefficients. Journal of Spacecraft and Rockets. https://doi.org/10.2514/1.46753

Crouch, E A.C.;Wilson, R. (1982). R. Risk/benefit analysis. Harvard University Press.

Fahimian, M., & Behdinan, K. (2017). On characterization of technology readiness level coefficients for design. Proceedings of the International Conference on Engineering Design, ICED.

GAO. (2015). Assessments of Selected Weapon Programs. Report to Congressional Committees.

GAO. (2018). James Webb Space Telescope: Integration and Test Challenges Have Delayed Launch and Threaten to Push Costs Over Cap.

Garg, T., Eppinger, S., Joglekar, N., & Olechowski, A. (2017). Using TRLS and system architecture to estimate technology integration risk. Proceedings of the International Conference on Engineering Design, ICED.

Hubbard, D., & Evans, D. (2010). Problems with scoring methods and ordinal scales in risk assessment. IBM Journal of Research and Development. https://doi.org/10.1147/JRD.2010.2042914

Ibrahim, D. (2016). An Overview of Soft Computing. Procedia Computer Science. https://doi.org/10.1016/j.procs.2016.09.366

Jackson, S. (2016). Systems Engineering for Commercial Aircraft. In Systems Engineering for Commercial Aircraft. https://doi.org/10.4324/9781315611723

Jaifer, R., Beauregard, Y., & Bhuiyan, N. (2020). New Framework For Effort And Time Drivers In Aerospace Product Development Projects. EMJ - Engineering Management Journal. https://doi.org/10.1080/10429247.2020.1772950

Kaplan, S., & Garrick, B. J. (1981). On The Quantitative Definition of Risk. Risk Analysis. https://doi.org/10.1111/j.1539-6924.1981.tb01350.x

Konar, A. (1999). Artificial Intelligence and Soft Computing. In Artificial Intelligence and Soft Computing. https://doi.org/10.1201/9781420049138

Kujawski, E. (2013). Analysis and critique of the system readiness level. IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans. https://doi.org/10.1109/TSMCA.2012.2209868

Lesinski, G. (2015). Application of Value Focused Thinking and Fuzzy Systems to Assess System Architecture. Procedia Computer Science. https://doi.org/10.1016/j.procs.2015.09.182

Magnaye, R. B., Sauser, B. J., & Ramirez-Marquez, J. E. (2010). System development planning using readiness levels in a cost of development minimization model. Systems Engineering. https://doi.org/10.1002/sys.20151

Mankins, J. C. (2009). Technology readiness and risk assessments: A new approach. Acta Astronautica. https://doi.org/10.1016/j.actaastro.2009.03.059

Marchetti, A. M. (2012). Overview of Enterprise Risk Management. In Enterprise Risk Management Best Practices. https://doi.org/10.1002/9781118386699.ch1

McConkie, E., Mazzuchi, T. A., Sarkani, S., & Marchette, D. (2013). Mathematical properties of System Readiness Levels. Systems Engineering. https://doi.org/10.1002/sys.21237

Newman, M. E. J. (2004). Detecting community structure in networks. The European Physical Journal B - Condensed Matter, 38(2), 321–330. https://doi.org/10.1140/epjb/e2004-00124-y

NIST. (2012). NIST Special Publication 800-30 Revision 1 - Guide for Conducting Risk Assessments. In NIST Special Publication.

Pizzuti, C. (2008). GA-Net: A genetic algorithm for community detection in social networks. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-540-87700-4_107

Project Management Institute, I. (2018). PMBOK Guide Sixth Edition. In Project Management Institute, Inc.

Reeves, J. D., Eveleigh, T., Holzer, T. H., & Sarkani, S. (2013). Risk identification biases and their impact to space system development project performance. EMJ - Engineering Management Journal. https://doi.org/10.1080/10429247.2013.11431970

Roedler, G. J., & Jones, C. (2005). Technical Measurement. A collaborative project of PSM, INCOSE, and industry. In INCOSE International Symposium.

Sauser, B., Gove, R., Forbes, E., & Ramirez-Marquez, J. E. (2010). Integration maturity metrics: Development of an integration readiness level. Information Knowledge Systems Management. https://doi.org/10.3233/iks-2010-0133

Sauser, B., Ramirez-marquez, J. E., & Tan, W. (2008). A Systems Approach to Expanding the Technology Readiness Level within Defense Acquisition. International Journal of Defense Acquisition Management.

Springsteen, Beth; Bailey, Elizabeth; Nash, Sarah; Woosley, J. (1999). Integrated Product and Process Development Case Study: Development of the F/A-18E/F. Retrieved from https://apps.dtic.mil/sti/pdfs/ADA379633.pdf

Thomas, P., Bratvold, R. B., & Bickel, J. E. (2014). The risk of using risk matrices. SPE Economics and Management. https://doi.org/10.2118/166269-pa

Yasseri, S., & Bahai, H. (2018). System Readiness Level Estimation of Oil and Gas Production Systems. International Journal of Coastal and Offshore Engineering. https://doi.org/10.29252/ijcoe.2.2.31

Yasseri, S. F., & Bahai, H. (2020). Case studies in estimating subsea systems' readiness level. Underwater Technology. https://doi.org/10.3723/ut.37.013

Zadeh, L. A. (1965). Fuzzy sets. Information and Control. https://doi.org/10.1016/S0019-9958(65)90241-X

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-I. Information Sciences. https://doi.org/10.1016/0020-0255(75)90036-5

SECTION

3. CONCLUSIONS AND RECOMMENDATIONS

One of the most important heuristic in systems engineering is to “simplify, simplify, simplify” everything we do to define, build, and test the system (Rechtin, E.; Maier, 2002). That is a powerful heuristic, though not so easy to do for complex systems to where we need to manage its complexity. Fundamentally, change in a complex network propagates between components through their interfaces and lack of interfaces can affect interconnectivty needs. Assessing complex interfaces and integration readiness is challenging yet a critical systems engineering skill because interface issues can produce system vulnerabilities (Walden, Roedler, & Forsberg, 2015).

Interface issues are not well explored in engineering literature and this dissertation makes a contribution to that area. Each chapter represents different methods to highlight interface and interconnectivty performance issues in complex systems that need to be exposed to systems managers, system developers, and decision makers to enable mitigation of system vulnerabilities due to these interfaces. The first paper uses a SoS Explorer (Version 2.1.0.1 Copyright© 2017 Missouri University of Science and Technology, Systems Engineering SMART Lab) to model, optimize, and visualize the heathcare system of systems architecture, allowing the architect to manipulate and evaluate the system to system interfaces against the overall healthcare capability and performance. This helps undertand the heathcare system of systems solution trade space and which interface and system to implement. The second paper provides a methodology to determine if the system or "community" of highly interconnected elements is ready to implement by analyzing maturity and performance of the elements and its interfaces within the system or community. A 777 business class seat system was used to prove out the methodology, where the seat system was partitioned into "communities" for interface analysis. Within these communities are highly interconnected elements that together as a cluster, key capabilities and performance characteristics are realized. Though, all the communities work together to enable the system's overall capability and measure of effectiveness. While the third paper does not address the need to calculate the overall capability of the communities based on performance and maturity of the system elemenets and interfaces, it provides a view of the integration risks of each community on a 5x5 risk matrix that is necessary for technical milestone reviews.

The soft-computing methodology to assess integraton readiness and performance of highly interconnected system elements and to quantify risks of each community for technical reviews is demonstrated in this dissertation. The second paper provides a cluster-based interface assessment methodology for breaking the system into a set of communities with strong interconnectivity for interface and performance analysis. Ultimately, understanding the system's network connectivity reduces bias from underestimating external interfaces by evaluating them with the interfaces established within a traditional hierarchical structure. The output of the methodology are community maturity level values and performance measures for each community to gage the level of integration risks which is the basis of the third paper, to quantify those risks based on these inputs. The third paper provides a risk assessment methodology that uses fuzzy principles to digest CML values and performance measures to quantify risks. The risks score of each community is placed on a 5x5 risk matrix for engjneering management reviews and decision-making.

Future work is to explore how certain communities when connected as a network in a system contribute to overall capability. There is opportunity to test the applicability of this methodology to other technical systems of varying scale and complexity such as residential and commercial power systems for a region or a large-scale software system. There is room for improving the CML equation that uses ordinal TRL and IRL values to consistently normalize the resulting values that allows us to use a risk matrix approach, such as using cardinal coefficients for TRLs and IRLs based on the Analytic Hierarchy Process (AHP). Furthermore, subject matter expert input on IRLs and performance measures are based on judgement and interpretation of data that may differ from other experts. There is work to do on validation of the inputs to better characterize readiness levels and performance that improve accuracy and quality of risk assessments. There is also validation work to do on fuzzy rules defined by subject matter experts to accurately characterize risks. Finally, there is opportunity to link DSM information to SysML diagrams used in MBSE applications and create an automated feedbackloop from CIAF analysis to SysML models.

BIBLIOGRAPHY

Ahsan, K; Hanifa, S; Kingston, P. (2010). Patients' Processes in Healthcare: an Abstract View Through Enterprise Architecture. International Conference on Information Management and Evaluation.

Al-Sakran, H. (2015). FRAMEWORK ARCHITECTURE FOR IMPROVING HEALTHCARE INFORMATION SYSTEMS USING AGENT TECHNOLOGY. International Journal of Managing Information Technology, 7(1). https://doi.org/10.5121/ijmit.2015.7102

Arena, M. V, Younossi, O., Brancato, K., Blickstein, I., & Grammich, C. A. (2008). Why has the cost of fixed-wing aircraft risen? A Macroscopic Examination of the Trends in U.S. Military Aircraft Costs over the Past Several Decades. In NATIONAL DEFENSE RESEARCH INSTITUTE and PROJECT AIR FORCE.

Beihoff, B., Friedenthal, S., Kemp, D., Nichols, D., Oster, C., Peredis, C., … Wade, J. (2014). A World In Motion: Systems Engineering Vision 2025. In Incose.

Boric, M. (2018, July). Electrification and E-Flight Part 4 Boeing Is on the Way to a (More) Electric Future. Aviation Pros.

Davies, P. (2020). Interface Management – the Neglected Orphan of Systems Engineering. INCOSE International Symposium, 30(1). https://doi.org/10.1002/j.2334-5837.2020.00752.x

Department of Defense Systems Management College. (2001). Systems Engineering Fundamentals - Supplementary Text Prepared by the Defense Acquisition University Press Fort Belvoir, Virginia. In Defense Acquisition University Press.

GAO. (2018). James Webb Space Telescope: Integration and Test Challenges Have Delayed Launch and Threaten to Push Costs Over Cap.

Herzlinger, R. E. (2006). Why innovation in health care is so hard. Harvard Business Review.

INCOSE. (2007). Systems Engineering Vision 2020. INCOSE-TP-2004-004-02. https://doi.org/citeulike-article-id:6888392

Jackson, S. (2016). Systems Engineering for Commercial Aircraft. In Systems Engineering for Commercial Aircraft. https://doi.org/10.4324/9781315611723

Jaifer, R., Beauregard, Y., & Bhuiyan, N. (2020). New Framework For Effort And Time Drivers In Aerospace Product Development Projects. EMJ - Engineering Management Journal. https://doi.org/10.1080/10429247.2020.1772950

Kaplan, G; Bo-Linn, G; Carayon, P; Pronovost, P; Rouse, W; Reid, P; Saunders, R. (2013). Bringing a Systems Approach to Health. Retrieved from https://doi.org/10.31478/201307a

Lu, B. (2010). The Boeing 787 Dreamliner: Designing an Aircraft for the Future. JOURNALS OF YOUNG INVESTIGATORS. Retrieved from https://www.jyi.org/2010-august/2010/8/6/the-boeing-787-dreamliner-designing-an-aircraft-for-the-future

Miles, A. (2009). Complexity in medicine and healthcare: people and systems, theory and practice. Journal of Evaluation in Clinical Practice, 15(3), 409–410. Retrieved from https://doi.org/10.1111/j.1365-2753.2009.01204.x

Muhammad, Sajid; Ahsan, K. (2016). ROLE OF ENTERPRISE ARCHITECTURE IN HEALTHCARE ORGANIZATIONS AND KNOWLEDGE-BASED MEDICAL DIAGNOSIS SYSTEM. Journal of Information Systems and Technology Management, 13(2). Retrieved from https://doi.org/10.4301/S1807-17752016000200002

Rechtin, E.; Maier, M. W. (2002). The Art of Systems Architecting. CRC Press.

Reeves, J. D., Eveleigh, T., Holzer, T. H., & Sarkani, S. (2013). Risk identification biases and their impact to space system development project performance. EMJ - Engineering Management Journal. https://doi.org/10.1080/10429247.2013.11431970

Rusnák, P. (2013). BOEING 787 DREAMLINER AIRCRAFT DEVELOPMENT AND OPERATION. ACTA AVIONICA, XV(27).

SoS Explorer Version 2.1.0.1 Copyright© 2017 Missouri University of Science and Technology, Systems Engineering SMART Lab

Walden, D. D., Roedler, G. J., & Forsberg, K. (2015). INCOSE Systems Engineering Handbook Version 4. INCOSE International Symposium, 25(1).

Watson, M. D. (2019). Future of Systems Engineering. INSIGHT, 22(1). https://doi.org/10.1002/inst.12231

VITA

Josh Henry Goldschmid had over fifteen years of experience in the aerospace industry as a structural analyst, an aircraft interior design engineer, a program manager, a systems engineer, and an engineering manager. He held a Bachelor of Science in Biological Systems Engineering at the University of California, Davis, a Masters of Science in Engineering Management at Drexel University, and has received his PhD in Systems Engineering at the Missouri University of Science and Technology in July 2021. Josh, as a profoundly Deaf person, was engaged as Technical Liaison for Disability:IN to elevate disability inclusion in the U.S. workplace, supply chain, and marketplace. Josh was a 2015 recipient of the Steven M. Atkins Ability and Achievement in Science, Engineering, and Technology (AASET) Award under the Society of Automotive Engineering (SAE) international organization.