VT-TE-18-229

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Tradespace Exploration for better Verification Strategies

5 September 2018

Dr. Alejandro Salado

Virginia Tech

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-17-1-0013. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

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

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

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

Abstract

This research, led by Dr. Alejandro Salado at Virginia Tech, addressed the definition of verification strategies in large-scale systems. Verification activities provide the evidence of contractual fulfillment. Thus, the importance of adequately defining verification activities in any acquisition program is unquestionable. Its significance extends beyond contracting though. The biggest portion of the development financial budget is spent in executing verification activities and verification activities are the main vehicle in discovering knowledge about the system, which is key to reduce development risk. In current practice, the definition of verification strategies is driven by industry standards and subject matter expert assessment. This approach leads to four major risks. First, there is a high uncertainty associated to the optimality of the selected verification strategy in terms of mitigated risk with respect to verification cost. Second, there is a lack of a quantitative risk measurement associated to chosen verification strategy, which jeopardizes any mindful effort to execute informed trade-off's regarding execution of verification activities. Third, there is a high risk associated to the verification coverage of the selected verification strategy, which threats the successful integration of components and the successful operation of the system. Fourth, there is a lack of alignment between stakeholder objectives and verification strategy, which leads to suboptimal decisions regarding the execution of verification activities.

In order to cope with these challenges, this research project addressed the main question of whether tradespace exploration can support the definition of more valuable verification strategies than current practice. In particular, the research had the following objectives: (1) To develop a metric for measuring the value of a verification activity, which will be achieved by elaborating a mathematical framework that computes the value of a verification activity as a function of the knowledge it discovers, and (2) To conduct a comparative analysis between tradespace exploration and a benchmark, which will be achieved by developing a tradespace exploration tool for verification strategies and use it to assess the value of verification strategies in actual, existing systems. The research employed a combination of

theoretical, mathematical elaborations and practical algorithm development and implementation. The hypotheses were tested on an Air Force Institute of Technology notional satellite.

By fulfilling the research objectives, this research is anticipated to significantly improve the value and cost of verification strategies. Furthermore, the direct public benefit of this research is anticipated to be higher early safety and efficacy of commercial products and public services. Finally, while we consider an application for the Air Force as a test case, we anticipate that the methodologies and insights provided in this work can be applicable to a broad range of systems that are subjected to limited verification: other defense systems, space systems, aeronautics, automotive systems, manufacturing systems, electronic products, civil infrastructure, public health systems, or transportation systems. The results of this research have been disseminated through professional conferences and scientific journals.

VT-TE-18-229

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Tradespace Exploration for better Verification Strategies

5 September 2018

Dr. Alejandro Salado

Virginia Tech

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-17-1-0013. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

THIS PAGE LEFT INTENTIONALLY BLANK

Table of Contents

Executive Summary 1

Background 3

Literature Investigation 7

Verification Engineering 7

Tradespace Exploration in Conceptual Design 8

Tradespace Exploration in Testing and Verification 10

Applying Tradespace Exploration to the Design of Verification Strategies 13

Framework 13

A mathematical model to generate verification strategies 14

A Bayesian Network Approach to Capture Information Dependencies 17

Valuation Metrics for Verification Strategies 20

Elicitation of Characteristics of Verification Activities 21

Case Study 25

Problem Overview 25

Benchmark Verification Strategy 26

Input Data 27

Tool Validation 30

Results 34

Recommendations and Future Directions 39

Conclusions 41

References 43

THIS PAGE LEFT INTENTIONALLY BLANK

List of Figures

Figure 1. Tradespace Exploration process applied to the design of a test strategy in (Salado, 2015) 11

Figure 2. Proposed Framework to Apply Tradespace Exploration to the Design of Verification Strategies 13

Figure 3. Simplified Firesat EPS Physical Hierarchy (Salado et al., 2018) 25

Figure 4. Base case verification strategy [from (Salado et al., 2018)] 27

Figure 5. Benchmark verification strategy as a Bayesian Network 31

Figure 6. Tradespace of verification strategies 35

THIS PAGE LEFT INTENTIONALLY BLANK

List of Tables

Table 1. Prior beliefs on SA, battery, and PCDU 27

Table 2. Prior belief for EPS.. 28

Table 3. Belief assignments for verification activities A, B, and C 28

Table 4. Belief assignment for verification activity D 29

Table 5. Belief assignment for verification activity E 29

Table 6. Belief assignment for verification activity F... 30

Table 7. Cost of verification activities 30

Table 8. Evolution of confidence on the system being absent of errors for Cases 1-3. 33

Table 9. Example of the value that verification activities provide as a function of prior knowledge 34

Table 10. Pareto front 36

THIS PAGE LEFT INTENTIONALLY BLANK

Executive Summary

This report describes recent research in support of acquisition programs facing the design of verification strategies in large-scale systems. Verification activities provide the evidence of contractual fulfillment. Thus, the importance of adequately defining verification activities in any acquisition program is unquestionable. Its significance extends beyond contracting though. The biggest portion of the development financial budget is spent in executing verification activities and verification activities are the main vehicle in discovering knowledge about the system, which is key to reduce development risk. In current practice, the definition of verification strategies is driven by industry standards and subject matter expert assessment. This approach leads to four major risks. First, there is a high uncertainty associated to the optimality of the selected verification strategy in terms of mitigated risk with respect to verification cost. Second, there is a lack of a quantitative risk measurement associated to chosen verification strategy, which jeopardizes any mindful effort to execute informed trade-off's regarding execution of verification activities. Third, there is a high risk associated to the verification coverage of the selected verification strategy, which threats the successful integration of components and the successful operation of the system. Fourth, there is a lack of alignment between stakeholder objectives and verification strategy, which leads to suboptimal decisions regarding the execution of verification activities.

In order to cope with these challenges, the research presented in this report addressed the main question of whether tradespace exploration could support the definition of more valuable verification strategies than current practice. In particular, the presented research had the following objectives: (1) Develop a metric for measuring the value of a verification activity, which was achieved by elaborating a mathematical framework that computes the value of a verification activity as a function of the knowledge it discovers, and (2) Conduct a comparative analysis between tradespace exploration and a benchmark, which was achieved by developing a tradespace exploration tool for verification strategies and using it to assess the value of verification strategies in a notional system. The presented

research employed a combination of theoretical, mathematical elaborations and practical algorithm development and implementation. The hypotheses have been tested on an Air Force Institute of Technology notional satellite.

This research is anticipated to significantly improve the value and cost of verification strategies. Furthermore, the direct public benefit of this research is anticipated to be higher early safety and efficacy of commercial products and public services. Finally, while an application for the Air Force has been considered as a test case, it is anticipated that the methodologies and insights provided in this work can be applicable to a broad range of systems that are subjected to limited verification: other defense systems, space systems, aeronautics, automotive systems, manufacturing systems, electronic products, civil infrastructure, public health systems, or transportation systems.

The research resulted in one published paper for the 2018 Naval Postgraduate School Acquisition Research Symposium, one published paper for the 2018 Conference on Systems Engineering Research (CSER), one published paper in the INCOSE's journal Systems Engineering, and one submitted paper to IEEE Systems Journal.

Background

Requirements are essential to system acquisition because they form the contractual vehicle by which customer/sponsor and contractor operate. For example, the contractor is expected to deliver to its customer a system that fulfills the agreed upon requirements. Demonstrating such fulfillment is achieved executing verification activities. Hence, the ultimate purpose of verification engineering is to provide evidence of contractual fulfillment. Thus, their importance for acquisition in contracting is unquestionable.

In this report, a verification strategy is understood to be a set of verification activities organized as an acyclic directed graph (Salado & Kannan, 2018b). A verification activity, which usually takes the form of analysis, inspection, or test, is understood to be the collection of information about a specific aspect of the system under development (for simplicity this will be called system parameter or requirement) and verification evidence refers to such information.

The significance of verification activities for acquisition extends beyond contracting though. Verification activities consume a significant part, if not the biggest part, of the development costs of large-scale engineered systems (Engel, 2010). Verification occurs at various system integration levels and at different times during the system's life cycle (Buede, 2009; Engel, 2010). For example, under a common master plan, low level verification activities are executed as risk mitigation activities, such as early identification of problems, or because some of them are not possible at higher levels of integration (Engel, 2010). In general, a verification strategy is defined "aiming at maximizing confidence on verification coverage, which facilitates convincing a customer that contractual obligations have been met; minimizing risk of undetected problems, which is important for a manufacturer's reputation and to ensure customer satisfaction once the system is operational; and minimizing invested effort, which is related to manufacturer's profit" (Salado, 2015). Essentially, verification activities are the main vehicle in discovering knowledge about the system and hence, they are key to reduce development risk. Because of this, defining valuable verification strategies is a key aspect of systems engineering management and test and evaluation in acquisition programs. In fact, verification strategies play a fundamental role in the affordability of large-scale systems, which continuous to be a great concern in government-funded projects as indicated by the General Accounting Office reports of the last 15 years.

In current practice, the design of verification strategies is driven by industry standards and subject matter expert assessment. Usually, the resulting strategy requires a cost higher than the initially budgeted by the project team. In these cases, de-scoping activities are performed, with qualitative evaluation of resulting risk, until agreement is reached by the engineering and project management teams. Such verification strategy is then agreed with the customer, following similar dynamics. Sometimes in parallel, but often after agreement with the customer, the prime contractor tries to impose its verification strategy to the lower level assemblies (developed by its subcontractors). This yields new negotiations and local trade-off's with each supplier. Same dynamics and approaches as described earlier are exhibited in these cases. Because the financial resources for such activities are usually committed at the early phases of a system's life cycle (Blanchard & Fabrycky, 1990; INCOSE, 2011), succeeding in finding an optimal strategy is often limited by the amount of time and resources that are invested in its definition, which are often scarce. In addition, because current practice relies on non-normative methods, the optimality of verification strategies currently defined in industry is questionable (Salado, 2015).

The context described above leads to four major risks. First, there is a high uncertainty associated to the optimality of the selected verification strategy in terms of mitigated risk with respect to verification cost. Second, there is a lack of a quantitative risk associated to the chosen verification strategy, which jeopardizes any mindful effort to execute informed trade-off's regarding execution of verification activities. Third, there is a high risk associated to the verification coverage of the selected verification strategy, which threats the successful integration of components and the successful operation of the system. Fourth, there is a lack of alignment between stakeholder objectives and verification strategy, which leads to suboptimal decisions regarding the execution of verification activities.

Informed by the benefits of tradespace exploration in conceptual design (Adam M. Ross & Hastings, 2005), the use of tradespace exploration was piloted in an actual industrial project to define a test strategy for a major satellite optical instrument with positive results (Salado, 2015). Specifically, using tradespace exploration yielded a test strategy with the same level of value and lower risk to the customer with 20% of lower cost than using the industry benchmark (Salado, 2015). However, such past work presented a number of limitations related to generality and normativity. The research project described in this report describes a generic framework that enables applying tradespace exploration to the design of verification strategies for any type of system. Specifically, the proposed approach to design verification strategies using tradespace exploration achieves the following objectives:

1) It enables quantitatively measuring the value of a verification activity and of a complete verification strategy. Such a measure facilitates the quantitative computation of risk and coverage.
2) It provides the necessary context to understand the achieved optimality of a verification strategy within a solution space.
3) It enables separating technical modeling from decision modeling, with facilitates aligning verification and stakeholder objectives.

THIS PAGE LEFT INTENTIONALLY BLANK

Literature Investigation

Verification Engineering

Verification of large-scale engineered systems may occur in every phase of their lifecycle (Engel, 2010), can take the form of a variety of methods (e.g. analysis, inspection, demonstration, test, or certification) (Engel, 2010), and can take place at different integration levels (INCOSE, 2011). Designing a verification strategy consists hence in deciding which verification activity occurs at which point in time and on what integration level. For example, method selection may be driven by programmatic constraints imposed by customers and business goals, credibility of method validity by customers, and feasibility of the method (Engel, 2010; Larson, Kirkpatrick, Sellers, Thomas, & Verma, 2009). Similarly, early verification, both in terms of assembly level and of lifecycle phase, may be desirable for mitigating the risk of failure or error (Engel, 2010; Firesmith, 2013) or because some system properties, attributes, or functionalities are not verifiable at higher levels of the assembly or cannot be verified in some specific configurations (Firesmith, 2013). Respectively, late testing may also be desirable for mitigating the risks of damage during the integration and test campaign and of emergent behavior or properties of all constituting elements integrated together (Firesmith, 2013), or simply because some system properties, attributes, or functionalities can only be verified once a number of elements are operating together (Firesmith, 2013).

In addition, designing a verification activity is driven by finding the right balance between verification cost and the cost of failure corresponding to those ones not discovered by the verification strategy (Engel, 2010). Since the cost and time allocated to verification activities represents a significant amount of the whole system development cost and time, optimizing verification is important in the development of large-scale systems (Engel & Shachar, 2006). Using cost and time as target values, several optimization techniques have been proposed as underlying mathematical/numerical model to identify a preferred verification strategy: loss function optimization, weight optimization, goal optimization, and genetic algorithm optimization (Engel, 2010; Engel & Shachar, 2006). Despite the diversity of methods though, all of them output a single optimum solution, i.e. they are point design strategies. Hence, they present the same limitations as point-design methods employed in conceptual design, as is described in the coming section.

Tradespace Exploration in Conceptual Design

Point-based design methods have been traditionally used in acquisition programs for the selection of conceptual or design alternatives. Such methods quickly anchor to a few solutions, which are then traded-off between themselves. Once selected, a given solution is further refined within the boundaries of its characteristics. The selection is therefore based on a relative comparison between the different alternatives, which can be understood as achieving a local optimum. Examples include the Analytical Hierarchy Process (AHP) (T. Saaty, 2004; T. L. Saaty, 1990); Pahl & Beitz's concept selection, where the best candidate resulting from calculating an overall score using utility theory and weighted average is chosen (Pahl & Beitz, 1996); fuzzy concept selection, where the best candidate resulting from calculating an overall score after using fuzzy logic on the selection criteria is chosen (Thurston, Carnahan, & Liu, 1994); or the concept selection method in flexible design strategies, where the best candidate resulting from calculating an overall score as a function of the individual system score and its concept flexibility score is chosen (King & Sivaloganathan, 1999). However, In contemporary system development, which is bounded by diverse stakeholders and multi-attribute decision making, traditional point design methods are found to be ineffective (Adam M. Ross & Hastings, 2005). Because the amount of alternative candidates is low, point design methods are unable to inform stakeholders about the *goodness* of a solution with respect to the broader solution space. Therefore, the question remains of whether the locally selected solution remains strong on a larger perspective. Only by looking at the entire (or a bigger portion) of the solution space this question can be answered with an adequate level of confidence.

As a response to such need, tradespace exploration techniques have been proposed (Adam M. Ross & Hastings, 2005). Using the concept of Pareto frontiers, these techniques recognize than in multi-attribute decisions a set of optimal solutions exists, as opposed to single optimum solutions. In this context, a Pareto frontier or front is a set of solutions that provide maximum return for a given level of investment (C. A. Mattson & Messac, 2003; A. M. Ross, Hastings, Warmkessel, & Diller, 2004). It should be noted that both the return and investment are not constrained to be of monetary nature, but they can also be accounted in non-monetary terms. Therefore, tradespace exploration techniques are based on populating a solution space first with as many solutions as possible, then identifying the set of optimal solutions (Pareto frontier), and finally letting stakeholders decide for a solution (Adam M. Ross & Hastings, 2005).

Tradespace exploration has been proven to support design methods that are effective in resolving ambiguity and facilitating communication, understanding, and agreement between multiple stakeholders (Golkar & Crawley, 2014; A. M. Ross et al., 2004). Furthermore, it also facilitates understanding underlying conflicting objectives during requirement elicitation and conceptual design (Salado & Nilchiani, 2015a, 2015b) and can support the planning of technology portfolios (Davison, Cameron, & Crawley, 2015). In addition, the capabilities of tradespace exploration are being steadily evolved to incorporate, for example, uncertainty in the determination of the Pareto front (C. Mattson & Messac, 2005), change over time in the tradespace configuration and location of the Pareto front (Curry & Ross, 2015; PK Lewis & Mattson, 2012; Adam M. Ross & Rhodes, 2008), value robustness (Adam M Ross, Rhodes, & Hastings, 2009), and framing and visualization effects for facilitating negotiation (Fitzgerald & Ross, 2014).

The diversity in domains where it has been used is growing. Positive results have already being obtained for selecting design concepts of manned and unmanned space systems (Golkar & Crawley, 2014; A. M. Ross et al., 2004; Shaw, Owens, Josan-Drinceanu, & Weck, 2014), aircrafts (PatrickK Lewis, Tackett, & Mattson, 2014), structural elements (PatrickK Lewis et al., 2014; C. Mattson & Messac, 2005), military operations (Hong, Wee, & Kiat, 2012), and Stirling engines (Smirnov & Golkar, 2015). In addition, tradespace exploration has been found effective in supporting concept selection of systems of systems both in research (Chattopadhyay, Ross, & Rhodes, 2008, 2009) and in practice (Salado, 2014). As a matter of fact, the method has now been infused as a standard practice for concept development at the Jet Propulsion Laboratory as part of their Concept Maturity Levels (Randii et al., 2013) and work is set forth for achieving common processes and tools at DoD (Spero, Avera, Valdez, & Goerger, 2014).

In addition, academia is also pursuing consolidation efforts and a research agenda has been proposed to advance the capabilities, maturity and infusion of tradespace exploration for conceptual design and system architecture (Spero, Bloebaum, German, Pyster, & Ross, 2014). In particular, four areas of work are anticipated:

(1) “Broadening, populating, and managing the tradespace”, which is primarily focused on characterizing and modeling the different elements that build up the tradespace, such as for example “ilities”, uncertainty, or historical data.

(2) “Linking the tradespace”, which is primarily focused on enabling technology compatibility, i.e. tradespace tooling.

(3) “Searching, exploring, and analyzing the tradespace”, which is primarily focused on improving the search process and the visualization mechanisms of tradespace exploration.

(4) “Acting on the tradespace”, which is primarily focused on harmonization activities in terms of foundations and processes.

The research agenda also discusses using tradespace exploration during the entire system life cycle. Yet, it explicitly understands this as updating the tradespace and resulting Pareto front of the system design or architecture as the system development evolves. The research agenda does not contemplate the potential value of tradespace exploration in systems engineering applications beyond conceptual design or system architecture. This report address such gap by exploring the use of tradespace exploration for designing verification strategies.

Tradespace Exploration in Testing and Verification

The application of tradespace exploration to the domain of verification was piloted in an industrial project to design the test strategy for a satellite instrument (Salado, 2015). The approach provided positive results, enabling the project team to uncover a test strategy that was less risky at 20% lower cost than the solution that was initially defined by the expert team using conventional definition approaches (Salado, 2015).Figure 1 shows the process that was developed for applying tradespace exploration in that project. Essentially, the processes starts with a test campaign that contains all potential test activities as described in (ECSS, 2012), which is then parsed into its elemental test activities. Such activities are then characterized in terms of cost and value to the customer, together with some general rules that account for couplings between the various activities. Finally, combinations of the different activities are generated to populate the solution space and evaluated.

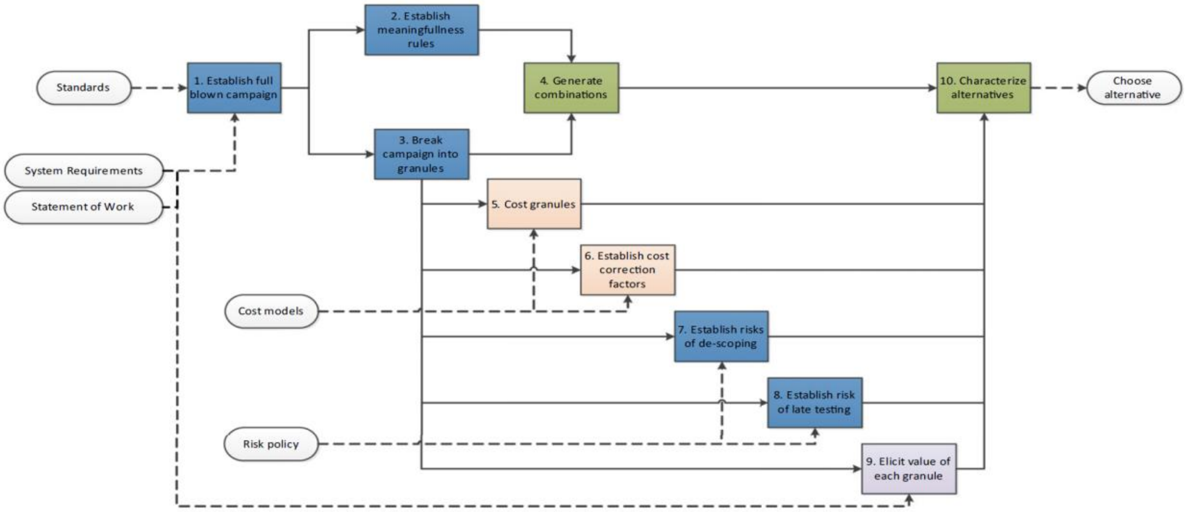
Standards
1. Establish full blown campaign
2. Establish meaningfullness rules
3. Break campaign into granules
4. Generate combinations
10. Characterize alternatives
Choose alternative
System Requirements
Statement of Work
5. Cost granules
6. Establish cost correction factors
Cost models
7. Establish risks of de-scoping
8. Establish risk of late testing
Risk policy
9. Elicit value of each granule

Figure 1. Tradespace Exploration process applied to the design of a test strategy in (Salado, 2015)

While the application of such process yielded positive results, the process had some limitations that disable it from being generalizable to other projects. Three limitations stand out:

1) The process was defined only for test activities and not verification activities in general. This implied that each activity was associated to a particular system characteristic. As a result, the process did not cover cases in which various verification activities are employed to build up together the verification evidence for a single system characteristic.
2) The sequence in which the test activities were to be executed was fixed. That is, the solution space only contained alternatives created by selecting which test activities would be performed, but only for a generic sequence. Therefore, a large portion of the solution space, containing different sequences of activities (Salado & Kannan, 2018b), was not explored.

3) Valuation of test strategies was qualitative and assumed a separable value function with respect to each test activity. As we will discuss later, valuing verification strategies is not straightforward and demands a more sophisticated approach that captures information dependencies between them (Salado & Kannan, 2018b, 2018c; Salado, Kannan, & Farkhondehmaal, 2018).

The work in (Salado, 2015) was expanded to overcome some of its limitations. In particular, mathematical foundations of verification engineering were proposed to enable the generalization of the application of tradespace exploration to defining verification strategies (Salado, 2016). Of particular importance is the realization that the purpose of verification activities is to discover knowledge about the system of interest (Salado, 2016). Consequently, the value of a given verification activity is not absolute (Salado & Kannan, 2018b). Instead, it is a function of the previous knowledge about the system of interest. Hence, the value of a verification activity depends, among others, on the verification activities that have been performed before it (Salado, 2016). This leads to two critical conclusions. First, sequence is a key driver of the value of verification strategies. Second, the value function for a verification activity may not easily be a separable function of its verification activities.

While the value of these dependency notions were showcased with a toy example, the mathematical foundations also presented some limitations that disable it from facilitating automation in the population of the solution space, as well as on adequately valuing verification strategies. In particular, the mathematical framework did not capture sequence of activities, although it was recognized in the sample case, and valuation was done qualitatively, without identifying a rigorous mathematical framework to enable computations. These limitations have been addressed in the project presented in this report.

Applying Tradespace Exploration to the Design of Verification Strategies

Framework

The developed framework builds upon the two main activities of tradespace exploration: generation of solutions and positioning of solutions in the tradespace. The framework is depicted in Figure 2. The generation activity consists in creating as many solutions as possible leveraging a structural model. The location activity consists in evaluating the generated solutions with respect to a set of predefined criteria, which would then result in positioning every solution within the tradespace.

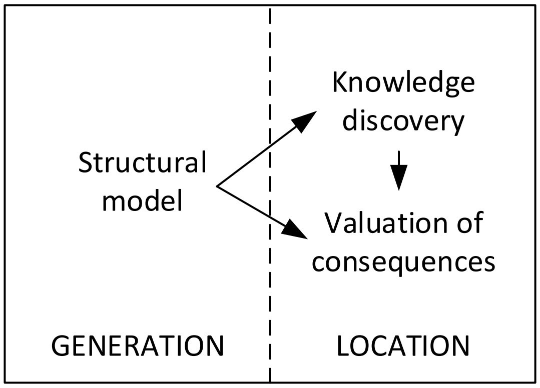

Figure 2. Proposed Framework to Apply Tradespace Exploration to the Design of Verification Strategies

The framework consist of three main elements, which are described in detail in the next sections:

1) A mathematical model that describes the underlying structure of verification strategies. This model enables automating the generation of verification strategies through computational algorithms to populate the solution space of verification strategies. The model is built with set theory and graph theory.
2) Mathematical machinery that, in combination with the structural model of verification strategies, enables one to compute the knowledge each verification strategy discovers. In other words, this element enables computing how verification strategies shape beliefs on the system

containing or not containing errors as verification activities are executed. This machinery is built using Bayesian networks.

3) A model that values the consequences of executing a verification strategy. In particular, expected value models are used to compute the cost associated to executing the verification strategy, as well as the expected cost to perform rework activities in case errors are actually found by a verification strategy.

A mathematical model to generate verification strategies

The mathematical model presented in this section is used to create the structures of the verification strategies that will fill the tradespace. First, it recognizes the existence of various verification activities and the notion that different activities may be used simultaneously to verify a single system characteristic. Second, it incorporates the capability to distinguish verification strategies as a function of the relative order in which verification activities are executed, not just their verification activities alone. Third, it does not impose any limitation on the valuation function in terms of separability. A full description of the model is given in (Salado & Kannan, 2018b).

The model is built on the following principles (Salado & Kannan, 2018b):

1) A verification strategy consists of verification activities executed and/or planned to be executed on a specific sequence[1]. This is distinguished from a verification plan in the sense that a plan incorporates specific programmatic elements to the foreseen execution of the verification strategy.

2) A verification activity provides information about the system on which the activity is executed, and consumes resources to be executed; these resources will be referred to as cost in this report. Therefore, the value of a verification activity depends on the information it provides and on its cost. As a result, the value of a verification activity is relative to preexisting information and resources at the time of its execution. Thus, the value of a verification activity depends on the information provided by previous verification activities, as well as to previous investments that could be potentially capitalized by the verification activity. Hence, two verification strategies are different if and only if their sets of verification activities are

[1] The term ‘sequence’ here should not be interpreted as the concept of *path* in a graph. Rather, the term sequence in this report refers to the order in which activities are executed, allowing parallel branches to exist. Essentially, following the form of any simple directed graph.

different or, if being the sets identical, the sequences in which they are executed are different.

3) Given a system that is decomposed in lower level components or parts (referred to as constituent elements in this report), information about the constituent elements is also relevant at a higher level. Therefore, the value of a verification activity at system level depends on the information provided by verification activities on its constituent elements. Consequently, the value of a verification activity also depends on the information provided by the verification activities on other same-level constituent elements. Hence, verification activities executed on constituent elements form part of a system's verification strategy.

4) Verification activities are executed to provide information about a finite set of system characteristics.

5) This report addresses only formal verification activities.

Let:

1) z_0 be the system of interest.

2) $z_1, ..., z_n$ be the systems that decompose z_0 in all of its constituent elements on which formal verification occurs. They are traditionally referred to as subsystems, components, or parts, among others.

3) $H_i = \left\{ z_i, z_{i,1}, z_{i,2}, ..., z_{i,m} \right\}$ be the set of systems that are homomorphic images of system z_i, as defined in (Wymore, 1993). Note that a system is homomorphic to itself and hence it is included in the set. This set represents all models of system z_i that are used for verification. In practical terms, they can take the form of a mathematical model, a prototype, or the final product, for example.

4) $F(z) = \left\{ p_1, p_2, ..., p_k \right\}$ be a parameterization of system z, where the definition of parameterization in (Wymore, 1993) is used. This parameterization is finite and represents the set of parameters of system z that need to be formally verified. For example, those parameters may represent the set of requirements that system z has to fulfill, and for which fulfillment needs to be proven through formal verification.

5) v be a **verification activity** defined as a tuple (p, r), where $p \in F(z)$ and $r \in R$. A verification activity is therefore understood as the application of a verification procedure r to the discovery of knowledge about a system parameter p.

Then, in order to capture the dependencies between verification activities, we define a **verification strategy** S as a simple directed graph $S = (V, D)$, where V is a set of verification activities and D is a set of tuples of the form (a,b) such that $a,b \in V$. The set V describes the verification activities that will be executed as part of S and D the relative order in which they will be executed. The solution space of verification strategies for a system z_0, denoted by $\Sigma(z_0,R)$, is given by all simple directed graphs that could be generated using all possible verification methods or procedures R on z_0. Mathematically, $\Sigma(z_0,R)=\{S=(V,D):V=\Upsilon(z_0,R)\}$, where $\Upsilon(z_0,R)$ is the set of all potential verification activities that could be executed to provide information about z_0. This is given by $\Upsilon(z_0,R)=\bigcup_{i=0}^{n}\left(F(z_i)\cup\bigcup_{j=1}^{|H_i|}F(z_{i,j})\right)\times R$.

Given a verification strategy S, multiple properties can be assigned to it. The main idea is to assign properties, such as information discovery or resources (e.g. cost, time, personnel, and so forth), to verification strategies so that they can be characterized. Given the existing dependencies between the nodes, as presented in this report, the properties cannot be simply decomposed into the nodes and edges of the strategy and rolled up together. Instead, the properties need to be assigned to subsets of the verification strategy that are independent of each other, but contain elements that are related within each subset by such dependencies. The presented model does not prescribe any specific set of properties. Instead, it provides a generic definition of property and a projection function to identify them. The mathematical definition follows.

Let $a=(a_1,a_2,...,a_n)$ be a finite list of properties a_1 through a_n, and define $g \circ f : S \rightarrow V \times A \rightarrow A$ to be a mapping between a verification strategy and a list of properties of the verification strategy. In particular, $f : S \rightarrow V \times A$ maps a verification strategy to lists of properties for each verification activity in the verification strategy and $g : V \times A \rightarrow A$ derives the properties of the verification strategy from the properties of the verification activities. So $g \circ f(S)$ is the list of properties associated with verification strategy S. It should be noted that the dependencies are not decomposed into independent properties of each verification activity. Instead, they are kept throughout the entire process because of the composition of both functions. That is, first the dependencies of the verification strategy are coded into $V \times A$, and then they are used when computing A. This is done by defining properties of verification activities as functions of properties of the verification activities to which they are connected through the edges in S.

A Bayesian Network Approach to Capture Information Dependencies

Bayesian networks (BN's) or belief networks are directed acyclic graphs that capture probabilistic dependencies and enable computing probabilistic or belief update using the Bayes' Theorem (Neapolitan, 2004). Essentially, a BN enables updating a prior belief on an event based on the arrival of new information. Structurally, a BN consists of nodes and arrows between the nodes. Nodes represent the random variables in which we are interested in gaining knowledge about and arrows between nodes represent the probabilistic dependencies between the nodes (Neapolitan, 2004).

As previously discussed, executing a verification activity may (and ideally should) result in updating the confidence in (uncertainty of) a system parameter – more generally, in an update of the confidence in the system being absent of errors (measured as a probability/belief). Hence, BN's seem to be adequate to model the information dependencies between verification activities in a verification strategy (Salado et al., 2018) and have been adopted for the framework presented in this report.

The following machinery is used (Salado & Kannan, 2018a). Let Θ be the set of all possible states of the system parameter θ that the engineer wishes to know (unobservable element). Let $S = \left\{V, D\right\}$ be a verification strategy intended to provide information about the system parameter θ. Let S^* be the set of all possible states of verification strategy S and denote by $\mathbf{s} \in S^*$ a specific vector of verification results (verification evidence; observable information). The use of *states* provides flexibility for capturing beliefs. For example, the state of a system parameter could be defined as the system being *compliant* or *not compliant* to a specific requirement that the system needs to fulfill, but it could also be defined as the exhibition of a particular system characteristic, such as the actual mass of the system being 340 kg. The same can be applied to the state of the results of a verification activity. It could be defined as the verification activity being considered *pass* or *no pass*, but it could also be the actual value indicated by the test, such as a measured mass of 339 kg.

A planned verification strategy is characterized as a density function $f\left(\mathbf{s}\,|\,\theta\right)$ for all $\mathbf{s} \in S^*$ and $\theta \in \Theta$, since its state depends on the system parameter θ on which information is sought. It should be noted that this function is applicable to every verification activity that yields information that could affect the belief about the state of the system parameter θ. A belief distribution of a parameter is denoted by $\pi\left(\cdot\right)$.

Hence, mathematically, the problem can be described as modeling the engineer's posterior belief distribution $\pi\left(\theta\,|\,\mathbf{s}\right)$ based on her prior belief distribution $\pi\left(\theta\right)$ and the density function $f\left(\mathbf{s}\,|\,\theta\right)$, conditioned on the collected verification evidence $\mathbf{s}$, which is given by $\pi\left(\theta\,|\,\mathbf{s}\right) = \frac{f\left(\mathbf{s}\,|\,\theta\right)\pi\left(\theta\right)}{\int_{\Theta} f\left(\mathbf{s}\,|\,\theta\right) dF^{\pi}\left(\theta\right)}$ as per the Bayes rule (Berger, 1985). The structural relationships are captured by an acyclic graph $S' = \left(\left\{\theta, V\right\}, D\right)$, where D is a set of vectors $\left(a,b\right)$ with $a,b \in \left\{\theta, V\right\}$, which indicates that a density function, $f\left(b\,|\,a\right) \neq f\left(b\right)$, exists. These structural relationships are the generalized to construct a model of the application of an entire verification strategy.

Consider a system z_0 built from components $z_1, \ldots, z_n$, for which verification models $z_1, \ldots, z_p$ are defined and which are characterized to a set of parameters $\theta_Z = \left\{\theta_1, ..., \theta_k\right\}$, and a verification strategy $\Upsilon = \left(V, D_I\right)$, where V is the set of verification activities $\left\{v_1, ..., v_m\right\}$ and D_I is the set of tuples that capture information dependencies between the various verification activities $\left\{\left(v_i, v_j\right), ..., \left(v_l, v_k\right)\right\}$, with $n, k, m, p \in \mathbb{N}$ and $v_i, v_j, v_l, v_k \in V$. Information dependencies are defined as those that exist between two verification activities, in the sense that the result of one of them affects the confidence gained or lost by executing the other one. For example, consider the verification activities v_1 to run a thermal dissipation analysis and v_2 to perform a thermal test. The tuple (v_1, v_2) indicates that the information provided by the thermal dissipation analysis influences our confidence in the success of the thermal test.

A Bayesian network that models the application of such verification strategy to such system can be constructed by combining three graphs. The first graph is the graph of the verification strategy, $\Upsilon = (V, D_I)$.

The second graph, denoted by $A = (\theta_Z, D_\theta)$, captures the properties of the system architecture. Specifically, it captures the prior beliefs on the absence of errors in the system parameters θ_Z, as well as the information dependencies between those parameters, given by $D_\theta = \{(a,b) : a, b \in \theta_Z, f(b \mid \mathbf{a}) \neq f(b)\}$. This graph essentially captures how information from system components provides insights about the system they build and vice versa. In other words, this graph captures the coupling between the different components building the system, as well as their individual maturity. For example, the mass of a system is a function of the mass of its components. Hence, our prior belief on the mass of the components shapes our prior belief on the mass of the system. Similarly, the prior belief on the mass of a new development may be different from the prior belief on the mass of a COTS system.

The third graph captures the ability of the verification activities to provide information about one or more system parameters. This graph is denoted by $B = (\{\theta_Z, V\}, D_\Upsilon)$, where $D_\Upsilon = \{(a,b) : a \in \theta_Z, b \in V, f(b \mid \mathbf{a}) \neq f(b)\}$. In testing, for example, this ability is commonly referred to as test specificity and sensitivity. However, the understanding of relevant information in this report takes a broader perspective, since the relevance is not a property of the verification activity alone, but also of the interpretation that the engineer makes out of the information yielded by the verification activity.

In summary, the resulting Bayesian network is given by $BN = \Upsilon \cup A \cup B$.

Valuation Metrics for Verification Strategies

Four value metrics for verification strategies are defined to characterize the tradespace.

Metric 1. The probability of the system exhibiting an error during operation, given that all verification activities were successful; (note that this type of error relates to malfunctioning, not derived from reliability). This metric is directly given by the Bayesian network described in the previous section.

Metric 2. The minimum cost associated to a verification strategy, that is, the cost of the verification strategy assuming that no error is found during the execution of the whole verification strategy. This metric is directly the investment necessary to execute the verification strategy. Simplistically, $c_{ex}(S) = \sum_{v \in S} c_{ex}(v)$, where $c_{ex}(v)$ is the cost of executing verification activity v.

Metric 3. The maximum cost associated to a verification strategy, that is, the cost of the verification strategy assuming that errors are found and corrected as late as possible. This metric is given by the investment necessary to execute the verification strategy and the cost of fixing all possible errors, which are identified on the last verification activity where they could be identified (in terms of sequence of activities).

Metric 4. The expected total cost of a verification strategy, which considers the possibilities of finding and correcting errors along the execution of the verification strategy. This metric is given by $E\left[c_{TOTAL}(S)\right] = E\left[c_{ex}(S)\right] + E\left[c_f(S)\right]$, where $E\left[c_f(S)\right]$ is the expected cost of fixing errors. For simplicity, it is assumed in the report that an error is fixed as soon as it is discovered and that a fixed error does not reemerge once it has been fixed. Under these conditions, the expected cost of fixing errors is given by $E\left[c_f(S)\right] = \sum_{i=1}^{\infty} \sum_{j=1}^{\#V} P\left(e_{i,j}\right) \cdot P\left(d_{i,j} \middle| e_{i,j}\right) \cdot c_f\left(e_{i,j}\right)$, where $P\left(e_{i,j}\right)$ is the probability that the system exhibits error e_i when verification activity v_j is executed, $P\left(d_{i,j} \middle| e_{i,j}\right)$ is the probability that verification activity v_j can discover error e_i (the discovery event is denoted by $d_{i,j}$), and $c_f\left(e_{i,j}\right)$ is the cost of fixing the error e_i when discovered by activity v_j. An error e_i will be exhibited by a system during the event v_j if at least one of two conditions is met. The first one is met when the error emerges after completion of v_{j-1} and before completion of v_j. The second one is met when the error has emerged earlier, but has not been discovered by previous verification activities. Hence, $P\left(e_{i,1}\right)=P\left(e_i \text{ em } 1\right)$ and

$$P\left(e_{i,j}\right)=\sum_{k=1}^{j-1}\left[P\left(e_i \text{ em } k\right)\cdot\prod_{l=1}^{k}\left(1-P\left(e_i \text{ em } l\text{ -}1\right)\right)\cdot\prod_{m=k}^{j-1}\left(1-P\left(d_{i,m}\middle|e_{i,m}\right)\right)\right]+P\left(e_i \text{ em } j\right)\cdot\prod_{k=1}^{j-1}\left(1-P\left(e_i \text{ em } k\right)\right)$$

, for $j \geq 2$, where $P\left(e_i \text{ em } j\right)$ is the probability that error e_i emerges after completion of v_{j-1} and before completion of v_j, and $P\left(e_i \text{ em } 0\right)=P\left(d_{i,0}\middle|e_{i,0}\right)=0$. The effect of the entire strategy is then incorporated by noting that the probability of an error being exhibited during a certain verification activity depends on its inherent nature of appearing at that point, as well as on the inability of the verification strategy to identify it earlier, if it emerged at an earlier point. It should be noted that these dependencies are defined by the Bayesian network presented in the previous section.

The four metrics can be combined in a common tradespace, where cost and probability of the system exhibiting an error are on the axes, and the different ranges of cost are shown with bars.

Elicitation of Characteristics of Verification Activities

The growth in computational demand of tradespace exploration as the number of solutions increases is a well-known problem. No effort has been done in the research project presented in this report related to optimizing the computational effort necessary to generate, characterize, and value solutions in the tradespace. However, the work has resulted in valuable insights regarding the effort necessary to elicit probabilities related to the performance of verification activities.

Three types of properties that require different approaches to characterize verification activities are identified:

1) *Stand-alone*: The value of the property for a verification activity is independent of any other verification activity in the verification strategy. Hence, the overall property for the verification strategy can be computed as a separable function of the properties of each verification activity. It is unwise to provide examples because most of them will be case specific.

2) *Rule-based*: The value of the property for a verification activity is defined as a set of cases, where each case is a function of the execution of other verification activities. The overall property for the verification strategy is computed by selecting the rules that apply, given its verification activities. Examples include, among others, reuse of verification equipment by various verification activities, which results in sharing the investment cost when applicable.

3) *Conditional probability-based*: The value of the property for a verification activity is a probability conditioned on the outcome of other verification activities. They are used in this work to characterize the probability of a requirement not met or, more generally, a parameter exhibiting certain behavior, as well as to characterize the results of verification activities on such parameters. These relationships can be defined via Conditional Probability Tables (CPT's) or through conditional probability functions. The work presented in this report is limited to CPT's.

Stand-alone properties need to be elicited just once for each verification activity. This is because, as stated earlier, their values will be the same for every verification strategy in the tradespace where the verification activity is executed. This is not the case for Rule-based and conditional probability-based properties.

Rule-based properties need to be elicited just once for each verification activity. Potentially, each verification activity may need up to one rule per each subset of the power set of the most comprehensive verification strategy, except the subset formed by the given verification activity. While the size of the problem can become very large easily, the nature of the rules in practical verification settings significantly reduces the complexity of the problem. Therefore, this aspect has not been furthered studied in this project.

Conditional probability tables need to be elicited just once for the most comprehensive verification strategy, that is, for the verification strategy that is a superset of every other verification strategy in the tradespace. This enables using the same probability values for every verification strategy in the tradespace, hence significantly reducing the elicitation effort for probabilities. This simplification is possible because, as a result of Bayesian inference, a verification strategy that does not include a given verification activity is equivalent to a verification strategy that includes such verification activity but does not execute it.

THIS PAGE LEFT INTENTIONALLY BLANK

Case Study

Problem Overview

The presented approach is showcased on a simplified version of Firesat's Electrical Power System (EPS). Firesat is a notional satellite that has been widely adopted for research in systems engineering (Wertz & Larson, 1999). The model captures the EPS as built of the following three components: the Power Control and Distribution Unit (PCDU), the Solar Panels (SA), and the Battery. A hierarchical breakdown of the system structure is depicted in Figure 3.

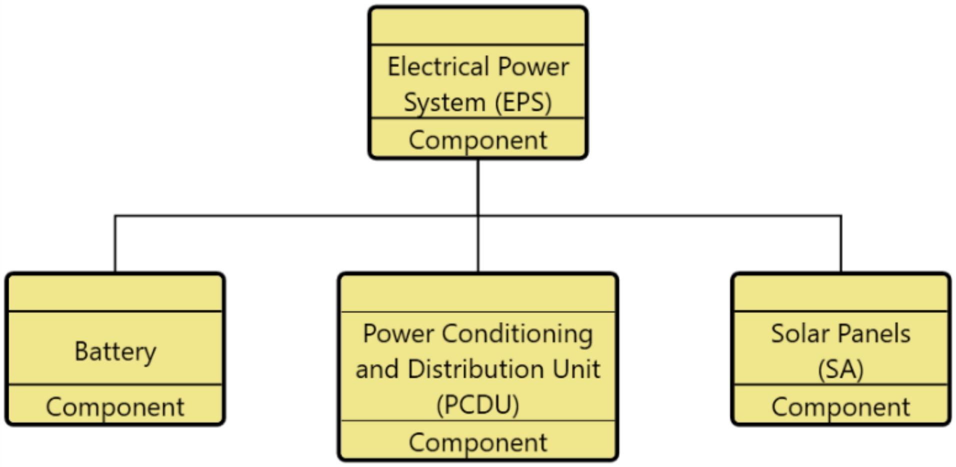
Electrical Power System (EPS)
Component
Battery
Component
Power Conditioning and Distribution Unit (PCDU)
Component
Solar Panels (SA)
Component

Figure 3. Simplified Firesat EPS Physical Hierarchy (Salado et al., 2018)

In order to investigate the impacts of various verification activities on the confidence on absence of error (i.e., proper functioning), the system model captures different levels of development maturity in the components that build the system (ECSS, 2009). Specifically, the following is assumed:

1) The EPS and PCDU need to be fully developed,
2) the SA is based on an existing unit but needs some modifications, and
3) the battery is recurring from a previous program.

Using varying levels of component maturity usually yields the need to use verification activities of varied fidelity levels. Hence, the notional EPS used in this case study enables a sensible definition of a notional verification strategy that incorporates diverse verification activities.

In addition, the following assumptions have been made to reduce the computational complexity of the case study, without affecting the generality of the application of tradespace exploration to the design of verification strategies:

1) It has been assumed that there is only one system characteristic that is verified and that verification can be achieved by analysis, test, or analysis and test on each building block in Figure 3.
2) Errors found during verification are not corrected.
3) Verification activities do not yield false positives.

Finally, synthetic data are used in this case study. Data have been defined under reasonable assumptions, in line with the maturity and coupling characteristics of the components defined previously. When operationalizing the approach presented in this report, such values may be elicited using existing estimation techniques, such as from subject matter experts or historical datasets. In any case, the nature of the data in this case study does not affect the purpose of this report, which is to display the application of tradespace exploration to design verification strategies.

Benchmark Verification Strategy

The notional verification strategy depicted in Figure 4 is used as a benchmark; arrows indicate temporal and information dependencies between the activities. The benchmark strategy includes the following activities: Analysis of SA (A), Analysis of Battery (B), Analysis of PCDU (C), Analysis of EPS (D), Test of PCDU (E), and Test of EPS (F).

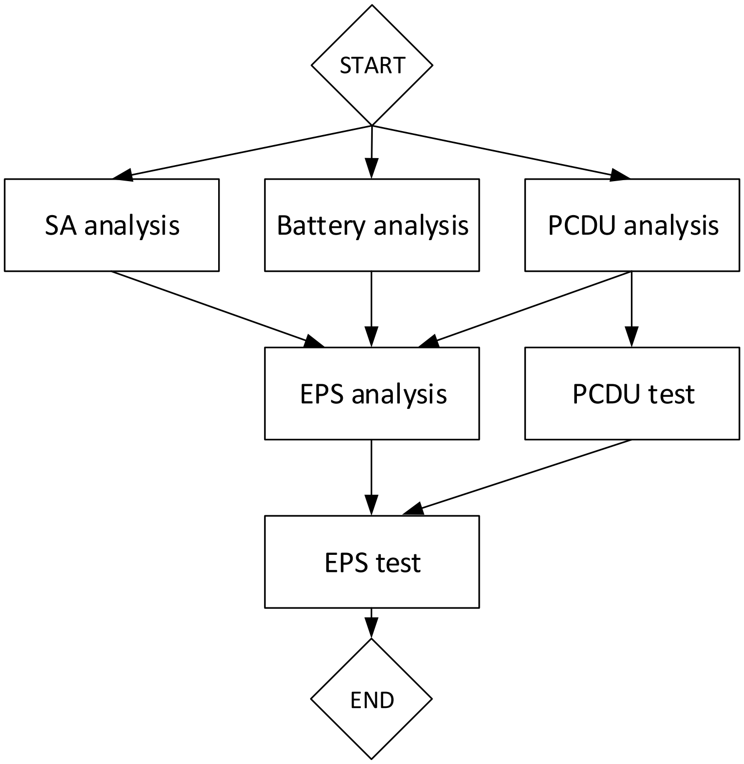

Figure 4. Base case verification strategy [from (Salado et al., 2018)]

While notional, the verification strategy is not arbitrary. Specifically, The level of verification fidelity has been defined as a function of the maturity of the components according to the guidelines in (ECSS, 2009).

Input Data

Table 1 lists the prior beliefs on the SA, the battery, and the PCDU exhibiting an error on parameters θ_{SA}, θ_{BAT}, and θ_{PCDU}, respectively. The existence of an error is denoted by e. Only those two states are considered, existence of an error and absence of an error. The beliefs of the components being absent of errors are the complements of the values in Table 1.

Table 1. Prior beliefs on SA, battery, and PCDU

Solar array	Battery	PCDU
$P(\theta_{SA}=e)=0.35$	$P(\theta_{BAT}=e)=0.05$	$P(\theta_{PCDU}=e)=0.50$

Table 2 lists the prior belief on the EPS exhibiting an error on parameter θ_{EPS}, conditioned to the SA, battery, and PCDU exhibiting or not exhibiting errors. The existence of an error is denoted by e and the absence of an error is denoted by $\neg e$. The belief on the EPS being absent of errors is the complement of the values in Table 2.

Table 2. Prior belief for EPS

θ_{SA}	θ_{BAT}	θ_{PCDU}	$P(\theta_{EPS} = \neg e \mid \theta_{SA}, \theta_{BAT}, \theta_{PCDU})$
$\neg e$	$\neg e$	$\neg e$	0.90
$\neg e$	$\neg e$	e	0.15
$\neg e$	e	$\neg e$	0.40
$\neg e$	e	e	0.15
e	$\neg e$	$\neg e$	0.40
e	$\neg e$	e	0.40
e	e	$\neg e$	0.40
e	e	e	0.10

Tables 3 through 6 list the beliefs assigned to the various verification activities conditioned to the components/system exhibiting errors and the results of previous verification activities with which information dependencies exist (as defined by Figure 4). Two states are considered for the results of the verification activity: pass and not passed, denoted by p and $\neg p$, respectively.

Table 3. Belief assignments for verification activities A, B, and C

A		B		C	
θ_{SA}	$P(A \mid \theta_{SA})$	θ_{BAT}	$P(B \mid \theta_{BAT})$	θ_{PCDU}	$P(C \mid \theta_{PCDU})$
$\neg e$	1.00	$\neg e$	1.00	$\neg e$	1.00
e	0.25	e	0.25	e	0.25

Table 4. Belief assignment for verification activity D

A	B	C	θ_{EPS}	$P\left(D = p \mid A, B, C, \theta_{EPS}\right)$
p	p	p	$\neg e$	1.00
p	p	p	e	0.40
p	p	$\neg p$	$\neg e$	1.00
p	p	$\neg p$	e	0.30
p	$\neg p$	p	$\neg e$	1.00
p	$\neg p$	p	e	0.30
p	$\neg p$	$\neg p$	$\neg e$	1.00
p	$\neg p$	$\neg p$	e	0.15
$\neg p$	p	p	$\neg e$	1.00
$\neg p$	p	p	e	0.30
$\neg p$	p	$\neg p$	$\neg e$	1.00
$\neg p$	p	$\neg p$	e	0.15
$\neg p$	$\neg p$	p	$\neg e$	1.00
$\neg p$	$\neg p$	p	e	0.15
$\neg p$	$\neg p$	$\neg p$	$\neg e$	1.00
$\neg p$	$\neg p$	$\neg p$	e	0.10

Table 5. Belief assignment for verification activity E

C	θ_{PCDU}	$P\left(E = p \mid C, \theta_{PCDU}\right)$
p	$\neg e$	1.00
p	e	0.30
$\neg p$	$\neg e$	1.00
$\neg p$	e	0.05

Table 6. Belief assignment for verification activity F

D	E	θ_{EPS}	$P\left(F=p \mid D,E,\theta_{EPS}\right)$
p	p	$\neg e$	1.00
p	p	e	0.20
p	$\neg p$	$\neg e$	1.00
p	$\neg p$	e	0.10
$\neg p$	p	$\neg e$	1.00
$\neg p$	p	e	0.10
$\neg p$	$\neg p$	$\neg e$	1.00
$\neg p$	$\neg p$	e	0.05

Table 7 lists the synthetic cost figures that have been used for each activity. It has been assumed that the cost of each verification activity is independent of each other for computational simplicity.

Table 7. Cost of verification activities

Verification activity	Cost
A	$100K
B	$200K
C	$300K
D	$500K
E	$800K
F	$1,000K

Tool Validation

The generation and evaluation of the tradespace was performed with a software program coded in Matlab© by the research team. Computation of Bayesian inference was validated by manual comparison of a few verification strategies modeled as Bayesian networks in the commercial software BayesServer©.

The same commercial software was employed to evaluate the meaningfulness of Bayesian models to capture the evolution of confidence on the system being free of errors as the results of verification activities become available. Figure 5 depicts the Bayesian Network (BN) that corresponds to the benchmark verification strategy. The nodes SA, Battery, PCDU, and EPS represent the prior belief on each component and the system exhibiting an error; that is, the confidence on each component and system properly functioning before any verification activity is carried out on them. Nodes A through F correspond to the verification activities presented in the previous section.

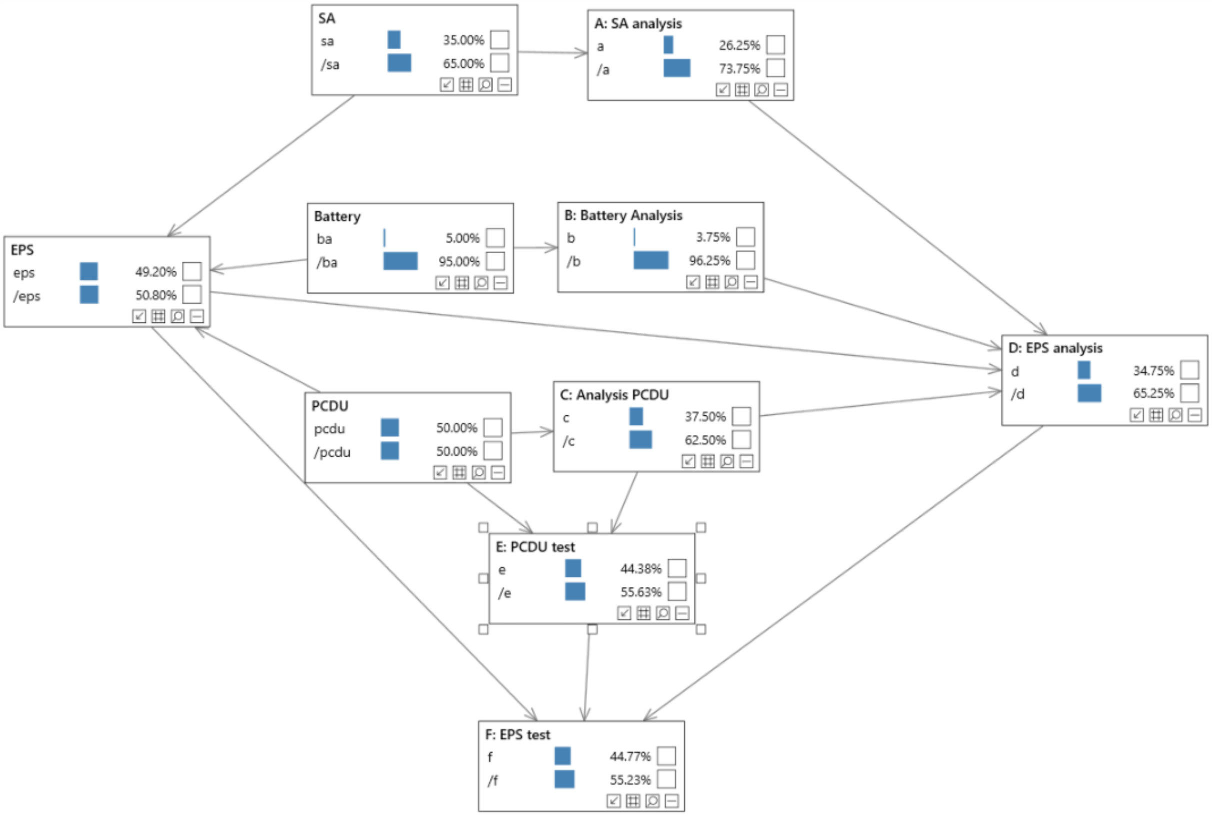

Figure 5. Benchmark verification strategy as a Bayesian Network

Four different cases of various predefined executions of the benchmark verification strategy were explored.

Case 1. It is assumed in this case that *all results are successful*. That is, no error is found in any verification activity after carrying out all verification activities in the strategy. Table 8 shows the evolution of the confidence on the system being absent of errors for this case. The BN indicates that the confidence of the system to function properly increases as successful results are confirmed. This behavior is consistent with how verification shapes confidence. There are two aspects worth mentioning. First, the impact of previous knowledge on the effect of verification activities should be noted. While an analysis on a mature component (the Battery in this case) results in marginal increase in confidence, an analysis of the same characteristics on an immature component (the PCDU in this case) yields a significant increase. Second, it is worth noting how the confidence on the proper functioning of the system increases thanks to verifying the system. Specifically with the values in this example, the notional verification strategy manages to increase it from 0.51 to 0.98.

Case 2. It has been assumed in this case that *all results are not successful*. That is, errors are found but not corrected in every verification activity after carrying out all verification activities in the strategy. Table 8 shows the evolution of the confidence on the system being absent of errors for this case. The BN indicates that the confidence of the system to function properly decreases as unsuccessful results are confirmed. This behavior is also consistent with how verification shapes confidence. Because it has been assumed that verification activities do not yield false positives, as well as no correction of errors, the strategy in this case reaches its end with activity D, since an error is found at the system level.

Case 3. A *combination of successful and unsuccessful results* has been assumed in this case. The combination has been defined arbitrarily. The results show that confidence updates as results become available, increasing with successful results and decreasing with unsuccessful ones. There are two aspects worth noting in this case. First, the Bayesian network captures the effects of margins at system level. While analyses have shown problems with the individual units, a successful result at system level indicates that they somehow compensate at system level. Yet, the final level of confidence that is achieved does not reach that one in Case 1.

Table 8. Evolution of confidence on the system being absent of errors for Cases 1-3.

Time	Case 1		Case 2		Case 3	
	v executed	1 – P(EPS)	*v* executed	1 – P(EPS)	*v* executed	1 – P(EPS)
T1	A: successful	0.59	A: unsuccessful	0.27	A: unsuccessful	0.27
T2	B: successful	0.60	B: unsuccessful	0.13	B: unsuccessful	0.13
T3	C: successful	0.74	C: unsuccessful	0.10	C: unsuccessful	0.10
T4	D: successful	0.88	D: unsuccessful	0.00	D: successful	0.53
T5	E: successful	0.91	n/a	n/a	E: successful	0.53
T6	F: successful	0.98	n/a	n/a	H: successful	0.92

Case 4. This case explores how the impact that a planned verification activity has on shaping the confidence on the system being absent of errors may change as results of prior verification activities become available. Two different sequences are defined. Other things being equal, in Sequence 1 activity E is performed before activity F. In Sequence 2, activity F is performed before activity E. That means that in Sequence 1 the results of activity E are known before executing activity F and vice versa in Sequence 2. Table 9 shows the BN prediction of how the confidence on the absence of errors in the EPS evolves in both cases. In Sequence 1, successful E increases the confidence from 0.88 to 0.91. However, the same activity in Sequence 2 only provides a marginal increase that is not even seeable with two-digit precision.

Hence, the BN shows that the *value* of verification activities cannot be measured in absolute terms, but is always conditioned to the knowledge available at the time of its execution.

Table 9. Example of the value that verification activities provide as a function of prior knowledge

Time	Sequence 1		Sequence 2	
	v	$1-P\left(\theta_{EPS}=\neg e\mid\forall v=p\right)$	v	$1-P\left(\theta_{EPS}=\neg e\mid\forall v=p\right)$
T	E, F not performed	0.88	E, F not performed	0.88
T+1	E: successful	0.91	F: successful	0.98
T+2	F: successful	0.98	E: successful	0.98

Results

The resulting tradespace is depicted in Figure 6. The Pareto front is listed in Table 10 and the benchmark strategy is the one on the top-right corner of the plot in Figure 6 for reference. It should also be noted that the elements in the Pareto front have been determined by rounding the probability values to two digits. This is the reason why, although the verification strategy consisting of all verification activities formally belongs to the Pareto front in this case (maximum probability of no error), it is finally not part of it because of the rounding effect. This assumption is reasonable because of the accuracy with which beliefs can be elicited.

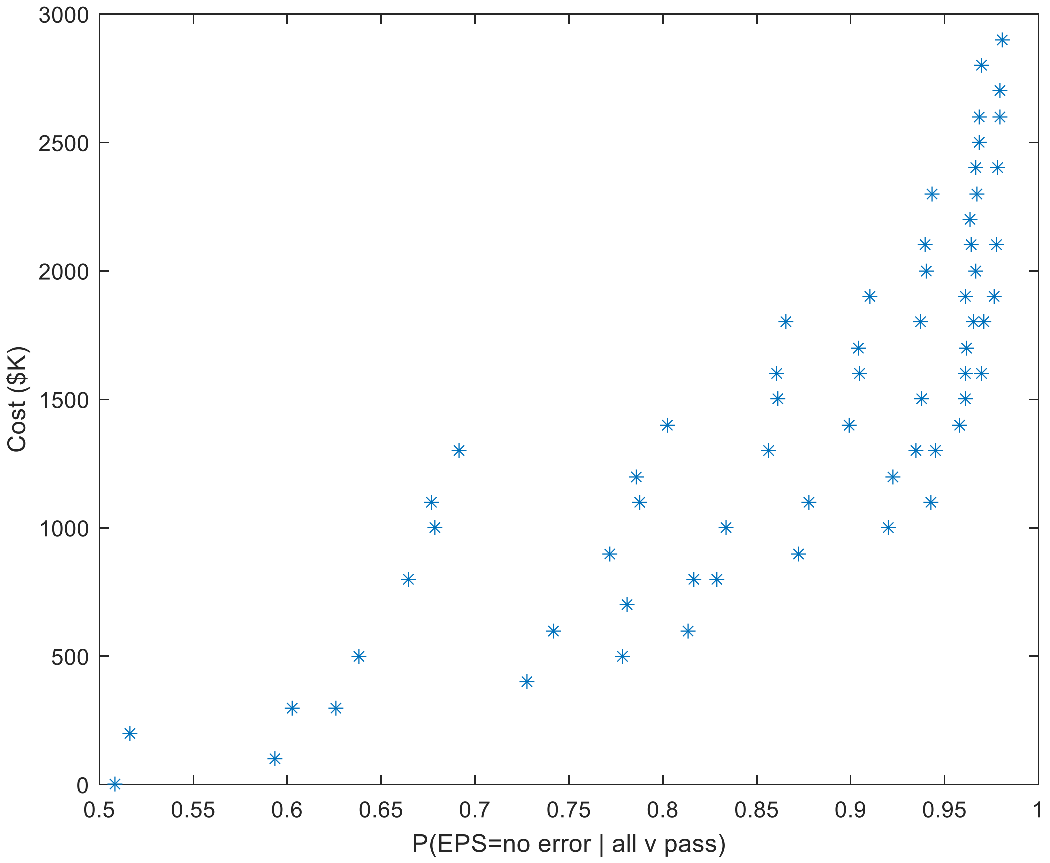

Figure 6. Tradespace of verification strategies

Table 10. Pareto front

Verification strategy	$P\left(\theta_{EPS} = \neg e \mid V = p\right)$	Cost
$V = \varnothing$	0.51	$0
$V = \{A\}$	0.59	$100K
$V = \{C\}$	0.63	$300K
$V = \{A, C\}$	0.73	$400K
$V = \{D\}$	0.78	$500K
$V = \{A, D\}$	0.81	$600K
$V = \{C, D\}$	0.83	$800K
$V = \{A, C, D\}$	0.87	$900K
$V = \{F\}$	0.92	$1,000K
$V = \{A, F\}$	0.94	$1,100K
$V = \{A, B, F\}$	0.95	$1,300K
$V = \{A, C, F\}$	0.96	$1,400K
$V = \{A, D, F\}$	0.97	$1,600K
$V = \{A, C, D, F\}$	0.98	$1,900K

*Note: Y indicates that the verification activity is part of the verification strategy and N means that the verification activity is not part of the verification strategy.

The two extremes in the plot are given by an empty verification strategy (that is, no verification activity is executed) and the one that consists of all verification activities in the base model. As expected, the empty verification strategy is free, but still yields a non-zero confidence on the correct functioning of the system, which is given by the prior beliefs on the components and the system itself. In addition, this type of analysis can provide information about the overall value of specific verification activities within the potential verification strategies for the given system development.

In this notional case, for example, verification activity E is not used in any verification strategy in the Pareto front. Similarly, verification activity A is employed in almost every verification strategy in the Pareto front.

In addition, as is the case with traditional tradespace exploration applied to concept selection, the evaluation of the tradespace in this case helps to understand how the different verification activities contribute to the value obtained by the overall verification strategy. The Pareto front serves the purpose to identify those verification strategies that are dominated, as well as to evaluate how much additional confidence is gained with delta increments or reductions in cost investments to execute additional verification activities.

THIS PAGE INTENTIONALLY LEFT BLANK

Recommendations and Future Directions

A key result of this report is that the necessity and value of a verification activity cannot be measured independently of the overall verification strategy. Instead, the necessity to perform a given verification activity depends on the results of all verification activities that have been previously performed. Essentially, the uncertain nature of system development will make verification activities that were not previously planned necessary, and will make some of the planned ones unnecessary. This implies that contractually committing to a fixed verification strategy at the beginning of an acquisition program fundamentally leads to suboptimal acquisition performance. Contrary to this finding, in current practice, a verification strategy is defined at the beginning of an acquisition program and is agreed upon by customer and contractor at contract signature. The tradespace exploration approach presented in this report can be extended to enable dynamic contracting of verification activities, which is necessary to guarantee optimality of acquisition programs in this area.

Dynamic evaluation of the tradespace has been explored using epoch-era analysis to consider how future scenarios can lead to changeability trade-off's in conceptual design and architecture (Curry & Ross, 2015; Rader, Ross, & Rhodes, 2010). It is suggested that a similar investigation is worth exploring for the approach presented in this report in the area of design of verification strategies. The initial approach presented in this report is still necessary to begin an acquisition program. In particular, it provides the expected confidence and cost of the verification program, as well as the bounds of the actual confidence that may be achieved and the cost that will be spent. Such information is critical to start planning dynamic contracting of verification activities.

Finally, the results presented in this report also suggest future work in the following areas to help operationalizing the application of tradespace exploration to the design of verification strategies:

1) Computational approaches that can efficiently handle the complexity of the problem of designing verification strategies.

2) Elicitation approaches at the intersection of expert opinion and historical data that can efficiently characterize prior beliefs and the performance of verification activities in a variety of verification strategies.

3) Reduction techniques that can predict

Conclusions

This report has presented a framework to apply tradespace exploration to the design of verification activities. The framework is built on mathematical machinery that enables the automated generation of verification strategies, the computation of the knowledge they discover, and the valuation of the consequences of executing them.

In the presented framework, a graph model provides the necessary structure to generate solutions that fill the tradespace. Bayesian analysis is employed to characterize the evolution the confidence an engineer's has on the system being absent of errors as the results of verification activities become available. Bayesian inference is used to reduce the effort necessary to elicit probabilities that characterize prior beliefs and the performance of the various verification activities in the context of the verification strategy to which they belong.

The presented framework overcomes the limitations of previous work. In particular, the framework recognizes the existence of various verification activities and the notion that different activities may be used simultaneously to verify a single system characteristic. Moreover, it is able to capture the dependencies between verification activities, enabling distinguishing verification strategies as a function of their sequences, not just of their verification activities. Furthermore, the presented framework does not impose any limitation on the valuation function in terms of separability.

The presented framework has been applied to a notional case study, specifically to a variant of the Firesat's Electrical Power System. The case study has demonstrated the feasibility to apply tradespace exploration to the design of verification strategies. In addition, it has shown that applying tradespace exploration can yield relevant information about the goodness of candidate verification strategies as it does when applied to conceptual selection.

The research presented in this report has also served the purpose to identify recommendations for future directions. Among them, probably the most relevant recommendation is to adopt dynamic contracting of verification activities, as opposed to agree on a fixed verification strategy at contract signature. The presented tradespace approach can be used, if provided with the capability to perform epoch-era analyses, to establish the dynamic evolution of the verification strategies that would yield optimal acquisition performance in this regard.

As a final note, the research presented in this report has resulted in one published paper for the 2018 Naval Postgraduate School Acquisition Research Symposium, one published paper for the 2018 Conference on Systems Engineering Research (CSER), one published paper in the INCOSE's Systems Engineering journal, and one submitted paper to IEEE Systems Journal, currently under review.

References

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis* (2nd ed.). New York, NY, USA: Springer Verlag.

Blanchard, B. S., & Fabrycky, W. J. (1990). *Systems engineering and analysis* (Vol. 4): Prentice Hall New Jersey;.

Buede, D. M. (2009). *The engineering design of systems: Models and methods*: Wiley.

Chattopadhyay, D., Ross, A. M., & Rhodes, D. H. (2008). *A Framework for Tradespace Exploration of Systems of Systems.* Paper presented at the Conference on Systems Engineering Research (CSER), Los Angeles, CA (USA).

Chattopadhyay, D., Ross, A. M., & Rhodes, D. H. (2009, 20-23 April 2009). *Combining Attributes for Systems of Systems in Multi-Attribute Tradespace Exploration.* Paper presented at the Conference on Systems Engineering Research (CSER), Loughborough University.

Curry, M. D., & Ross, A. M. (2015). Considerations for an Extended Framework for Interactive Epoch-Era Analysis. *Procedia Computer Science, 44*(0), 454-465. doi:http://dx.doi.org/10.1016/j.procs.2015.03.029

Davison, P., Cameron, B., & Crawley, E. F. (2015). Technology Portfolio Planning by Weighted Graph Analysis of System Architectures. *Systems Engineering, 18*(1), 45-58. doi:10.1002/sys.21287

ECSS. (2009). Space engineering - Verification. Noordwijk, The Netherlands: European Cooperation for Space Standardization.

ECSS. (2012). Space engineering - Testing. Noordwijk, The Netherlands: European Cooperation for Space Standardization.

Engel, A. (2010). *Verification, Validation, and Testing of Engineered Systems*. Hoboken, NJ: John Wiley & Sons, Inc.

Engel, A., & Shachar, S. (2006). Measuring and optimizing systems' quality costs and project duration. *Systems Engineering, 9*(3), 259-280. doi:10.1002/sys.20056

Firesmith, D. (2013). *Common Testing Pitfalls and Ways to Prevent and Mitigate Them: Descriptions, Symptoms, Consequences, Causes, and Recommendations*: Addison Wesley.

Fitzgerald, M. E., & Ross, A. M. (2014). Controlling for Framing Effects in Multi-stakeholder Tradespace Exploration. *Procedia Computer Science, 28*(0), 412-421. doi:http://dx.doi.org/10.1016/j.procs.2014.03.051

Golkar, A., & Crawley, E. F. (2014). A Framework for Space Systems Architecture under Stakeholder Objectives Ambiguity. *Systems Engineering, 17*(4), 479-502. doi:10.1111/sys.21286

Hong, L. J., Wee, T. C., & Kiat, L. Y. (2012, 28-31 October 2012). *Tradespace Exploration for Military Simulations.* Paper presented at the 2012 Autumn Simulation Multiconference, San Diego, CA (USA).

INCOSE. (2011). *INCOSE Systems engineering handbook, v.3.2.2*. Retrieved from San Diego, CA:

King, A. M., & Sivaloganathan, S. (1999). Development of a methodology for concept selection in flexible design strategies. *Journal of Engineering Design, 10*(4), 329-349.

Larson, W. J., Kirkpatrick, D., Sellers, J. J., Thomas, D., & Verma, D. (2009). *Applied space systems engineering*: McGraw Hills.

Lewis, P., & Mattson, C. (2012). A method for developing systems that traverse the Pareto frontiers of multiple system concepts through modularity. *Structural and Multidisciplinary Optimization, 45*(4), 467-478.

Lewis, P., Tackett, M. P., & Mattson, C. (2014). Considering dynamic Pareto frontiers in decision making. *Optimization and Engineering, 15*(4), 837-854. doi:10.1007/s11081-013-9238-2

Mattson, C., & Messac, A. (2005). Pareto Frontier Based Concept Selection Under Uncertainty, with Visualization. *Optimization and Engineering, 6*(1), 85-115. doi:10.1023/B:OPTE.0000048538.35456.45

Mattson, C. A., & Messac, A. (2003). Concept Selection Using s-Pareto Frontiers. *AIAA Journal, 41*(6), 1190-1198.

Neapolitan, R. E. (2004). *Learning Bayesian Networks*. Upper Saddle River, NJ, USA: Pearson Prentice Hall.

Pahl, G., & Beitz, W. (1996). *Engineering design - A systematic approach*. London: Springer.

Rader, A. A., Ross, A. M., & Rhodes, D. H. (2010, 5-8 April 2010). *A methodological comparison of Monte Carlo Simulation and Epoch-Era Analysis for tradespace exploration in an uncertain environment.* Paper presented at the 2010 IEEE International Systems Conference.

Randii, W., Chester, S. B., John, K. Z., Robert, C. M., Joan, E., & Jared, L. (2013). Space Mission Concept Development using Concept Maturity Levels *AIAA SPACE 2013 Conference and Exposition*: American Institute of Aeronautics and Astronautics.

Ross, A. M., & Hastings, D. E. (2005). 11.4.3 The Tradespace Exploration Paradigm. *INCOSE International Symposium, 15*(1), 1706-1718. doi:10.1002/j.2334-5837.2005.tb00783.x

Ross, A. M., Hastings, D. E., Warmkessel, J. M., & Diller, N. P. (2004). Multi-attribute tradespace exploration as front end for effective space system design. *Journal of Spacecraft and Rockets, 41*(1), 20-28.

Ross, A. M., & Rhodes, D. H. (2008). 11.1.1 Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis. *INCOSE International Symposium, 18*(1), 1186-1201. doi:10.1002/j.2334-5837.2008.tb00871.x

Ross, A. M., Rhodes, D. H., & Hastings, D. E. (2009). *Using pareto trace to determine system passive value robustness.* Paper presented at the Systems Conference, 2009 3rd Annual IEEE.

Saaty, T. (2004). Decision making — the Analytic Hierarchy and Network Processes (AHP/ANP). *Journal of Systems Science and Systems Engineering, 13*(1), 1-35. doi:10.1007/s11518-006-0151-5

Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European journal of operational research, 48*(1), 9-26.

Salado, A. (2014). *An Effective Approach to Explore Conceptual Solutions in Systems of Systems.* Paper presented at the 6th International Systems & Concurrent Engineering for Space Applications Conference (SECESA 2014), Vaihingen Campus, University of Stuttgart, Stuttgart, Germany.

Salado, A. (2015). Defining Better Test Strategies with Tradespace Exploration Techniques and Pareto Fronts: Application in an Industrial Project. *Systems Engineering, 18*(6), 639-658. doi:10.1002/sys.21332

Salado, A. (2016). *Applying tradespace exploration to verification engineering: From practice to theory and back again.* Paper presented at the Conference on Systems Engineering Research (CSER), Huntsville, AL (USA).

Salado, A., & Kannan, H. (2018a). Elemental Patterns of Verification Strategies. *Systems Engineering, Under review*.

Salado, A., & Kannan, H. (2018b). A mathematical model of verification strategies. *Systems Engineering, In press*.

Salado, A., & Kannan, H. (2018c). *Properties of the Utility of Verification*. Paper presented at the IEEE International Symposium in Systems Engineering, Rome, Italy.

Salado, A., Kannan, H., & Farkhondehmaal, F. (2018). *Capturing the Information Dependencies of Verification Activities with Bayesian Networks*. Paper presented at the Conference on Systems Engineering Research (CSER), Charlottesville, VA, USA.

Salado, A., & Nilchiani, R. (2015a). Affordability Benefits of Fractionated Spacecraft in Certain Futures: A Case Study on Decoupling Conflicting Requirements. *Under review*.

Salado, A., & Nilchiani, R. (2015b). The Tension Matrix and the Concept of Elemental Decomposition: Improving Identification of Conflicting Requirements. *Systems Journal, IEEE, PP*(99), 1-12. doi:10.1109/JSYST.2015.2423658

Shaw, M. M., Owens, A. C., Josan-Drinceanu, I., & Weck, O. L. d. (2014). *Multidisciplinary Hybrid Surface Habitat Tradespace Exploration and Optimization*.

Smirnov, D., & Golkar, A. (2015). Stirling Engine Systems Tradespace Exploration Framework. *Procedia Computer Science, 44*(0), 558-567. doi:http://dx.doi.org/10.1016/j.procs.2015.03.010

Spero, E., Avera, M. P., Valdez, P. E., & Goerger, S. R. (2014). Tradespace Exploration for the Engineering of Resilient Systems. *Procedia Computer Science, 28*(0), 591-600. doi:http://dx.doi.org/10.1016/j.procs.2014.03.072

Spero, E., Bloebaum, C. L., German, B. J., Pyster, A., & Ross, A. M. (2014). A Research Agenda for Tradespace Exploration and Analysis of Engineered Resilient Systems. *Procedia Computer Science, 28*(0), 763-772. doi:http://dx.doi.org/10.1016/j.procs.2014.03.091

Thurston, D. L., Carnahan, J. V., & Liu, T. (1994). Optimization of design utility. *Journal of Mechanical Design, 116*(3), 801-808.

Wertz, J. R., & Larson, W. J. (1999). *Space mission analysis and design*: Microcosm.

Wymore, A. W. (1993). *Model-based systems engineering*. Boca Raton, FL: CRC Press.

Acquisition Research Program
Graduate School of Business & Public Policy
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net