SYM-AM-18-105

PROCEEDINGS
OF THE
FIFTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

THURSDAY SESSIONS
VOLUME II

Acquisition Research:
Creating Synergy for Informed Change

May 9–10, 2018

March 30, 2018

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

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

A Mathematical Framework to Apply Tradespace Exploration to the Design of Verification Strategies

Alejandro Salado—is an Assistant Professor with the Grado Department of Industrial and Systems Engineering at Virginia Tech. His research focuses on applying decision analysis to improve the practice of engineering, in particular in the areas of verification and validation. He is a recipient of the Fulbright International Science and Technology Award. Dr. Salado holds a BSc/MSc in electrical engineering (Polytechnic University of Valencia), an MSc in project management, and an MSc in electronics engineering (Polytechnic University of Catalonia), the SpaceTech MEng in space systems engineering (Delft University of Technology), and a PhD in systems engineering (Stevens Institute of Technology). [asalado@vt.edu]

Farshad Farkhondehmaal—is a PhD student of management systems engineering with the Grado Department of Industrial and Systems Engineering at Virginia Tech. He received his BSc in aerospace engineering from Sharif University of Technology in Iran. His research interests are systems engineering, model-based systems engineering, and systems dynamic. [farshad1@vt.edu]

Abstract

This paper is intended to disseminate some initial outcomes of the NPS Research Acquisition Program's "Tradespace Exploration for Better Verification Strategies" project. The research addresses 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 research addresses the main question of whether tradespace exploration can support the definition of more valuable verification strategies than current practice. We present in this paper a mathematical framework that enables the application of tradespace exploration to the design of verification strategies.

Introduction

Requirements lay at the core of system acquisition, given their contractual nature. Verification activities are executed to demonstrate fulfillment of those requirements. Hence, verification provides the evidence of contractual fulfillment. Actually, in several cases reaching an agreement about when a requirement is considered fulfilled is more important than agreeing on the requirement itself. Thus, their importance for acquisition in contracting is unquestionable.

Verification activities, which may take the form of a combination of analyses, inspections, and tests, consume a significant part, if not the biggest part, of the development costs of large-scale engineered systems (Engel, 2010). Verification occurs at various levels of a system's decomposition and at different times during its life cycle (Buede, 2009; Engel, 2010). 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). Therefore, a verification strategy is defined as

> 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 vehicle by which contractors can collect evidence of contractual fulfillment in acquisition programs.

In current practice, the definition of verification strategies is driven by industry standards and subject matter expert assessment. Usually, the resulting strategy requires a higher cost than the initial budget allocated by the project. De-scoping activities are then performed, with qualitative evaluation of resulting risk, until agreement is reached by the engineering and project management teams. Such verification strategy is then agreed on with the customer, following similar dynamics. Sometimes in parallel, but often after agreement with the customer, the prime contractor tries then to impose its verification strategy to the lower level assemblies (developed by its subcontractors). This yields new negotiations and local trade-offs with each supplier. The 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 (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.

Furthermore, current practice relies on non-normative methods that are based on subject matter expert assessments rather than on measurements, which questions the optimality of verification strategies currently defined in industry (Salado, 2015). This context 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 chosen verification strategy, which jeopardizes any mindful effort to execute informed trade-offs regarding execution of verification activities. Third, there is a high risk associated to the verification coverage of the selected verification strategy, which threatens the successful integration of components and the successful operation of the system. And 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 (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 (Salado, 2015). The results were positive, being able to identify a test strategy with the same level of value and lower risk to the customer with a 20% lower cost than using the industry benchmark (Salado, 2015). However, the work presented a number of limitations related to generality and normativity. This paper presents a modified framework to apply tradespace exploration to the design of verification strategies that overcomes those limitations.

Background

Tradespace Exploration

Traditional point design methods have been found to be ineffective in traditional concept design (Ross & Hastings, 2005). Such methods quickly anchor to a few solutions, limiting the perspective on potentially better solutions available in the larger solution space. As a response to such need, tradespace exploration techniques have been proposed (Ross & Hastings, 2005). They recognize that in multi-attribute decisions, a set of optimal solutions exists, as opposed to a single optimum solution. In this context, tradespace exploration consists in comprehensively populate the solution space with as many solutions as possible, identify its Pareto frontier or front (which is a set of solutions that provide maximum return for a given level of investment), and let the stakeholder choose a solution (Mattson & Messac, 2003; Ross & Hastings, 2005; Ross et al., 2004). 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; Ross et al., 2004).

Verification in Large-Scale Engineered Systems

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 of 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, et al., 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 models 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, which have been identified in the previous section.

Tradespace Exploration Applied to the Verification Domain

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 *Space Engineering—Testing* (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.

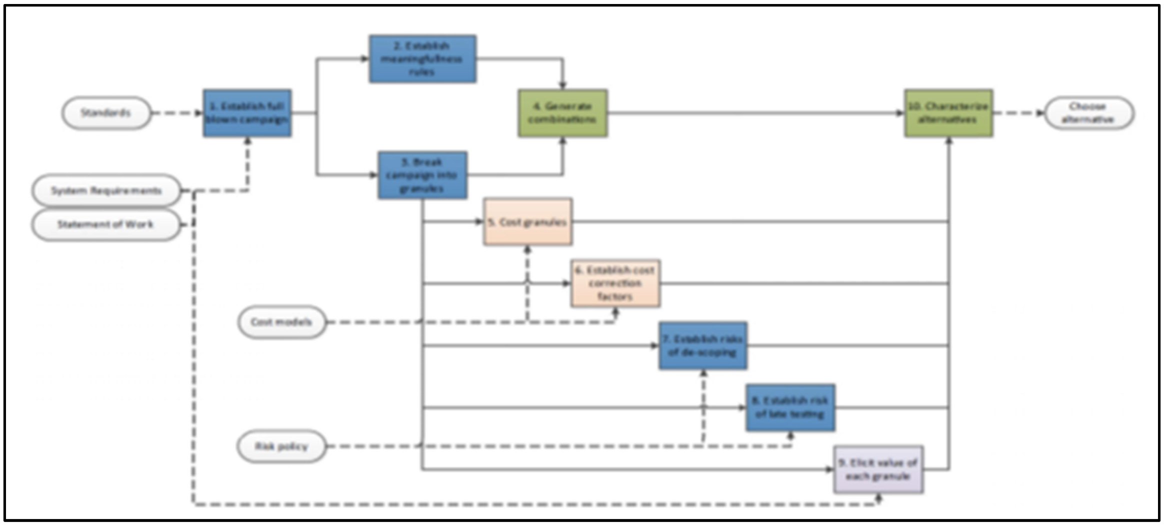

Figure 1. Tradespace Exploration Process Applied to the Design of a Test Strategy
(Salado, 2015)

While the application of such a process yielded positive results, the process had some limitations that disable it from being generalizable to other projects. Three limitations stand out. First, 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. Second, 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 verification activities would be performed, but only for a generic sequence. Therefore, a large portion of the solution space, containing different sequences of activities, was not explored. Third, valuation of verification strategies was qualitative and assumed a separable value function with respect to each verification activity. As we will discuss later, valuing verification strategies is not straightforward and demands a more sophisticated approach.

Salado's (2015) work 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. Instead, it is a function of the previous knowledge about the system of interest. Hence, the value 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 present 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.

In this paper, we present a comprehensive framework that overcomes all limitations of previous work in the application of tradespace exploration to the design of verification activities.

A Tradespace Exploration Framework to Design Verification Strategies

Framework

We propose a framework that 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 of 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.

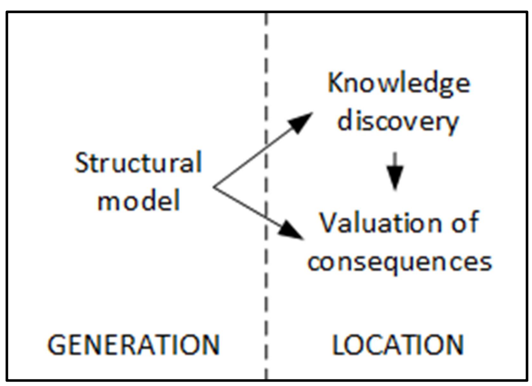

Figure 2. Proposed Framework to Apply Tradespace Exploration to the Design of Verification Strategies

The framework consist of three main elements, which are described in detail in the next sections. First, we make use of 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. Second, we add machinery to the structural model of verification strategies to compute the knowledge they discovered. In other words, how verification strategies shape beliefs on the system containing or not containing errors as verification activities are executed. This machinery is built on Bayesian networks. Third, we valuate the consequences of executing a verification strategy. In particular, we provide expected value models to compute the cost associated with 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

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 $a, b \in V$. Then, V describes the verification activities that will be executed as part of S and D the sequence in which they will be executed. The solution space of verification strategies for a system z_0, denoted by $\sum(z_0, R)$, will therefore be given by all simple directed graphs that could be generated using all possible verification methods or procedures R on z_0. Mathematically, $\sum(z_0, R) = \{S = V, D): V = \Upsilon(z_0, R)\}$, 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

$$\Upsilon(z_0, R) = \bigcup_{i=0}^{n}\left(F(z_i) \cup \bigcup_{j=1}^{\#H_i} F(z_{i,j})\right) \times R,$$

where

1. $z_1, \ldots, z_n$ are 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.
2. $H_i = \{z_i, z_{i,1}, z_{i,2,} \ldots, z_{i,m}\}$ is 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.
3. $F(z) = \{p_1, p_2, \ldots, p_k\}$ is 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.
4. A **verification activity** v is 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.

This mathematical framework overcomes the limitations of previous work. 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 their sequences, not just their verification activities. Third, it does not impose any limitation on the valuation function in terms of separability. This will be shown later in the Valuing a Verification Strategy section.

A Bayesian Network to Capture Information Dependencies

Since the mathematical construct for describing a verification strategy presented in the previous section is a directed graph, it enables seamlessly embedding a Bayesian network to enable calculations related to beliefs or probabilities of errors existing in the system being developed and of the verification activities discovering those errors. We define now the Bayesian network machinery, as applied to model the knowledge discovery of a verification strategy. A detailed description of how to create the Bayesian network is given in Salado, Kannan, and Farkhondehmaal (2018). A summary follows.

Using the mathematical model presented in the previous section, consider a system z_0 built from components $z_{1,1}, \dots, z_{1,n}$ and a verification strategy $S = (V, D)$, where V is the set of verification activities $v_1, \dots, v_m$ and D is the set of tuples that capture information dependencies between the various verification activities $\{(v_i, v_j), \dots, (v_l, v_k)\}$, with $n, m \in$ and $v_i, v_j, v_l, v_k \in V$.

A Bayesian network that models such verification strategy can be constructed by combining three graphs. The first one contains directly the graph of the verification strategy, $S_1 = (V, D)$. The second graph the Bayesian networks contains captures the prior belief on the absence of errors in the various components that form the system and the system itself and the first verification activities executed on them. Mathematically, we can denote such graph as $I = (Z, A)$, where $Z = \{z_0, z_{1,1}, \dots, z_{1,n}\}$ and $A = \{v \in V : \exists z \in Z \text{ such that } P(v_{pass}|z) \neq P(v_{pass})\}$ and the outcomes of each verification activity v can only be v_{pass} and $\neg v_{pass}$. This graph captures the dependency between the prior knowledge about the components forming the system, including the system itself, and the first verification activities that are carried out in the verification strategy. Finally, the Bayesian network must contain the belief on the absence of error on the system z_0 as it relates to the belief on its components being absent of errors. Mathematically, we can denote such graph as $F = (Z, B)$, where $B = \{(z_i, z_j) : z_i, z_j \in V, P(z_i|z_j) \neq P(z_i)\}$. This graph captures the coupling between the different components forming the system, that is, how they inform the confidence on the proper functioning of the system.

In summary, the resulting Bayesian network is given by $BN = (V \cup Z, D \cup A \cup B)$.

Valuing a Verification Strategy

We have defined four value metrics for verification strategies:

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).
2. The minimum cost associated to the verification strategy, that is, the cost of the verification strategy assuming that no error is found during the execution of the whole verification strategy.
3. The maximum cost associated to the verification strategy, that is, the cost of the verification strategy assuming that errors are found and corrected as late as possible.
4. The expected cost associated to the verification strategy, which considers the possibilities of finding and correcting errors along the execution of the verification strategy.

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. Now we define the four metrics mathematically.

Metric 1. The probability of the system exhibiting an error during operation, given that all verification activities were successful, is directly given by the Bayesian network described in the previous section. Hence, no further description is needed.

Metric 2. The minimum cost associated to a verification strategy 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 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 is given by $E[c_{TOTAL}(S)] = E[c_{ex}(S)] + E[c_f(S)]$, where $E[c_f(S)]$ is the expected cost of fixing errors. We assume in this paper 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, we define $E[c_f(S)] = \sum_{i=1}^{\infty} \sum_{j=1}^{\#V} P(e_{i,j}) \cdot P(d_{i,j} | e_{i,j}) \cdot c_f(e_{i,j})$, where $P(e_{i,j})$ is the probability that the system exhibits error e_i when verification activity v_j is executed, $P(d_{i,j} | e_{i,j})$ is the probability that verification activity v_j can discover error e_i (the discovery event is denoted by $d_{i,j}$), and $c_f(e_{i,j})$ 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, $P(e_{i,1}) = P(e_i \text{ em } 1)$ and

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

for $j \geq 2$, where $P(e_i \text{ em } j)$ is the probability that error e_i emerges after completion of v_{j-1} and before completion of v_j , and $P(e_i \text{ em } 0) = P(d_{i,0} | e_{i,0}) = 0$. 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.

Challenges in Implementing the Framework to a Sample Case

Overview of the Problem

The application of the presented framework to a sample case is an ongoing effort. While the results will be presented at a different venue, we discuss in this section the challenges associated to operationalizing the mathematical framework presented in this paper.

The sample case in Salado et al. (2018) is used as a starting point. The system of interest is a simplified version of the Electric Power System (EPS) of the FireSat satellite (Wertz & Larson, 1999). A hierarchical breakdown of the system structure is depicted in Figure 3. The system model captures different levels of development maturity in the components that build the system (ECSS, 2009). Specifically, it is assumed that the EPS and PCDU need to be fully developed, the SA is based on an existing unit but needs some modifications, and the battery is recurring from a previous program.

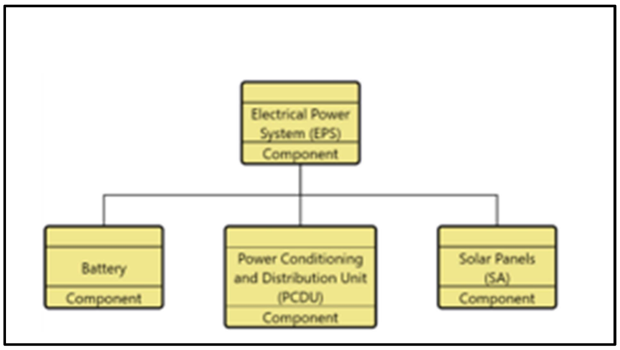

Figure 3. Simplified Firesat EPS Physical Hierarchy
(Salado et al., 2018)

For simplicity, we also assume 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.

Process

As discussed in the previous section, we start to populate the tradespace by applying the combinatorial effort to a fixed pattern of sequences of verification activities. This is done to limit the necessary computational effort, in particular in terms of eliciting conditional probabilities. We use as a base case the notional verification strategy defined in Salado et al. (2018), which is depicted in Figure 4. It reflects the order in which verification activities are executed (from top to bottom), as well as the information dependencies between them in the form of arrows. The verification strategy is defined as generic, without targeting any specific system characteristic.

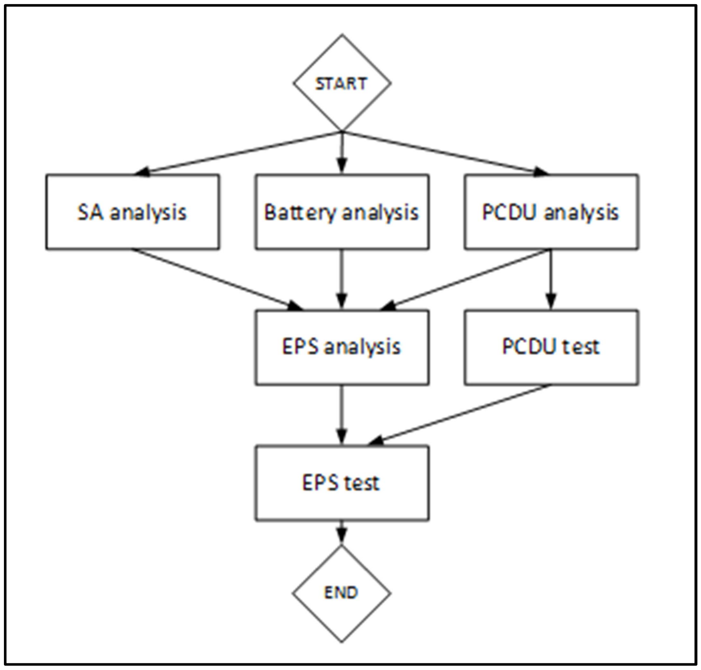

Figure 4. Base Case Verification Strategy
(Salado et al., 2018)

Therefore, the first tradespace is formed by all verification strategies that can be formed by applying or not applying each of the verification activities in Figure 4. Then, we incorporate a relative sequence change, leading to the modified base strategy and incorporate to the tradespace all verification strategies resulting from the combinations of applying or not applying each of the verification activities. Finally, we incorporate another relative sequence change, leading to a second modified base strategy and repeat the operation.

Valuation of verification strategies is also performed in two stages, in order to limit computational effort. First, each verification strategy in the tradespace is characterized by its minimum cost and the knowledge they discover. As previously described, minimum cost is defined as the execution cost of the strategy and knowledge discovery is defined as the probability of the system exhibiting once operational after all the verification activities in the sequence have been successful. Second, a subset of verification strategies that provide a similar knowledge discovery is selected and their expected and maximum costs, which depend on the actual finding of errors and exercise of repair costs, are calculated, forming a new tradespace.

Challenges and Way Forward

Operationalizing the presented mathematical framework into a computational code has presented some challenges. Two deserve particular attention.

The first one, inherent to tradespace exploration, is the exponential growth of the tradespace with verification activities and system characteristics. However, problem size becomes larger and more intricate due to the dependencies between the activities. In particular, it should be noted that given a set of n possible verification activities, there are 2^n sets of verification activities, and $4^{\binom{n}{2}}$ directed graphs for each one of the sets of verification activities. In addition to the common methods employed in tradespace exploration to reduce the size of the problem, identifying in advance the independence between knowledge generation helps in identifying equivalent sequences, thus reducing the size of the problem.

The second issue is related to the availability of conditional probability tables for each verification strategy. Conditional probability tables depend heavily on the specific sequences and reusing and adjusting them seems to be non-trivial. In a worst-case scenario, a dedicated set of conditional probability tables would need to be created for each verification strategy in the tradespace. Of course, this is infeasible. Furthermore, the nature of conditional probabilities in verification strategies makes it difficult to create a model that can be used to automate the generation of conditional probability tables. This problem can be overcome with a sufficiently large database of historical performance of verification strategies executed on systems similar to the system of interest. Given the lack of a publicly available database of such kind, we are currently developing a synthetic database of verification strategies for Earth observation satellites to support this study.

Conclusions

We have presented in this paper a framework to apply tradespace exploration to the design of verification activities. The framework is built on mathematical machinery that enable the automated generation of verification strategies, the computation of the knowledge they discover, and the valuation of the consequences of executing them.

The proposed frame.work overcomes the limitations of previous work. In particular, the proposed 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 their verification activities. Furthermore, the proposed framework does not impose any limitation on the valuation function in terms of separability.

Finally, we have discussed the challenges that we are finding when operationalizing the mathematical framework to apply it to a sample case. The effort is ongoing and is planned to be completed within the timeframe of the NPS Research Acquisition Program's "Tradespace Exploration for Better Verification Strategies" project.

References

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

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.

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*. Boston, MA: Addison Wesley.

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

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

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

Mattson, C. A., & Messac, A. (2003). Concept selection using s-pareto prontiers. *AIAA Journal, 41*(6), 1190–1198.

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.

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.

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.

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.

Acknowledgments

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.

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

www.acquisitionresearch.net