Exploring A Verification Complexity Framework

Sukhwan Jung
The University of Arizona
1127 E James E Rogers Way
Tucson, AZ, USA
shjung@arizona.edu

Alejandro Salado
The University of Arizona
1127 E James E Rogers Way
Tucson, AZ, USA
alejandrosalado@arizona.edu

Abstract. While system complexity is considered an integral piece of information throughout the system development life cycle, the complexity of the verification is given comparatively lower focus in the field of systems engineering. There is no domain-wide consensus on the definition of verification complexity, resulting in disputed complexity measures or lack thereof. Verification is a pervasive task throughout the system development; its insufficient measurement is detrimental to both the system engineers and users. We propose the Verification Complexity Framework as a formal definition of verification complexity. A cube-shaped framework is proposed to cover both static and dynamic complexity through the time axis and the hierarchical complexity layers, covering from external effects to the verification structures. Its modular design allows the framework to be nested to mimic information flow between verification at multiple integration levels. This framework provides a common vocabulary for verification complexity, where both its definition and measurements can be discussed.

Introduction

System complexity impacts project outcomes, affecting the time, resources, and cognitive power required for successful development (Sheard & Mostashari 2013). Verification strategies are part of system development and defining their complexity could allow better impact prediction for the V&V process. The complexity of verification is, however, a rarely studied topic despite its potential importance throughout the system lifecycle (Jung & Salado 2023). As a starting point of this research field, we aim at providing a common vocabulary to complexity with regards to verification as opposed to the system being verified. The prevalent use of verification processes in system development renders this research field valuable in both theoretical and practical applications. While some researchers distinguish between complexity and complicatedness as visualized in the Cynefin framework (Snowden 2010), we use the term *complexity* to describe both in this paper.

We propose the Verification Complexity Framework (VCF) to promote discussions on the definition of verification complexity. Its structure is evaluated within the context of systems engineering. In the VCF, layers of verification complexity are defined by a systemic review of existing research as well as experiences in industrial projects. The experience-based tacit knowledge within the verification engineering communities is formalized while maintaining consistency with the philosophies of system complexity frameworks. Considerations to mathematical quantifications are undergoing to facilitate measurement and practical usability in the future (Jung & Salado 2024). These measures aim to provide quantifications to each complexity layer, promoting the VCF as a unified complexity management tool used throughout the system lifecycle.

The VCF is designed to measure complexity throughout the system life cycle, tracking system verification throughout the design, implementation, and operational stages. We satisfied this life cycle trackability by implementing two specific capabilities. Firstly, the VCF is modular, so it can be nested to represent the verification complexity derived from the system architecture. Multiple verification activities occurring across system integration levels can be represented as individual VCFs interconnected to each other. The overall verification complexity over the whole system could be calculated by integrating the verification complexity of individual VCFs traversing the interconnections. The VCF has standardized input and output functions allowing a directional interconnection. Secondly, the VCF incorporates time as a dimension. This allows a dynamic representation (or evolution) of complexity as it maps a system development.

This paper is organized as follows. Background discusses the background research on system and verification complexities. The VCF and its Layers section describes the proposed framework. The Conclusion section provides discussions and concluding remarks on the proposed framework and future works that are being conducted and planned.

Background

System Complexity. Researchers do not have a consensus on the definition of system complexity, as the definition of the system itself is complex to begin with. The working definitions range from uncertainty in system and activity, interrelation dynamics, information processing requirements, generation of emergent behaviors (Weaver 1948), to cyclomatic tendency (Sheard & Mostashari 2013) and structural complexity (Sinha & de Weck 2016). Subjective complexity such as representation, perception, and human factors have been also utilized (Fässberg et al. 2011). The Systems Engineering Body of Knowledge (Adcock, Sillitto & Sheard 2023) defined system complexity under three categories: structural, dynamic, and socio-political (Sheard & Mostashari 2011). The socio-political category includes environmental factors such as regulations and politics overarching the product and project complexities in the context of project management (Botchkarev & Finnigan 2015). Another three-categorical framework was proposed using self-control, redundancy, and repeatability (Brooks & Roy 2020) noting distinctions between static and dynamic system complexity properties (Efatmaneshnik & Ryan 2016).

System modularity, on which intrinsic system complexity is distributed over the elements, was linked to the integrative complexity of the system (Sinha, Suh & de Weck 2018). Problem complexity was proposed to provide insight into the complexity caused by the problem instead of an already existing system (Salado & Nilchiani 2014). Assessing multiple complexity types and symptoms was found to be necessary, but industry does not have a consensus on a unified approach (Alkan et al. 2018). Sometimes these complexities are simplified by analogical application of traditional mathematical entities. For example, information entropy has been utilized to represent information uncertainties in system operations (Zhang 2012), while the Reynold number was used to illustrate manufacturing flow turbulences (Romano 2009). The idea of information and flow within the system resulted in the use of graph-based complexity measures such as degree and centrality (Alkan et al. 2016). Within this variety of definitions, the size of the system is often linked to the system complexity in practice as a simplistic measure, counting system elements for example (Schoettl, Paefgen & Lindemann 2014).

Verification Complexity. Verification complexity is a loosely defined term with various interpretations proposed over the years. Some approach the verification problem by tying verification complexity to the problem characteristics such as implementation and specification (Harel, Kupferman & Vardi 2002). Verification complexity can also refer to a computational complexity with mathematical principles with the research focusing on determining the problem space completeness of automata systems and their observability (Rohloff & Lafortune 2005). More recent research represented verification complexity by diagnosability, predictability, and detectability (Yin & Lafortune 2017).

Another research focused on human factors, calculating verification complexity by counting the number of information processing tasks required by the verifying user (Lyell & Coiera 2017).

Ranging from interview-based qualitative figures to computational formulas, different measures and definitions for verification complexity have been proposed. In most cases, verification complexity, albeit having multiple facets of meaning, is directly linked to the complexity of the system being verified. The correlation between system complexity and verification complexity seems an accepted phenomenon (Linehan & Clarke 2010), which is still used in recent years (Horváth et al. 2023). Essentially, verification is complex when the system being verified is complex. We use the term 'verification complexity' in a general sense, aiming to encompass various definitions in the VCF.

Verification Strategy. Verification is commonly performed to check that *"the system was built right"* by comparing the system design, system models, and the system itself against the required characteristics or requirements (Buede & Miller 2016; Walden 2015). A verification action is taken for any element matching the result against the predefined answer set. Such elements can take a variety of forms, including for example system architecture, design, implementation, and performance (Laing et al. 2020). They are also generally planned and carried out throughout the system life cycle parallel to the system development (ISO 2023). Verification may be performed through multiple methods such as inspection, analysis, analogy, sampling, or testing (Engel 2010; Wibben & Furfaro 2015). Both manual and automatic methods can be employed. Each verification activity needs to be designed with practical considerations; external parameters such as available equipment, proficiency of relevant personnel, and contractual/regulatory constraints need to be considered.

These verification activities for individual requirements are then combined in a specific sequence to form a verification strategy for the whole system. These are designed to reach a certain confidence level that *the system is built right* with the correct functionalities. The loose definition of verification can lead to the explosion of verification activities, therefore resource control has become an integral part of the verification strategy design minimizing the invested resources (Engel 2010). This requires importance categorization of the verifiable properties as well as constraint management. Due to the wide range of internal and external considerations, designing verification strategies often relies on the qualitative experience of subject matter expert opinions and industry standards (e.g. (ESCC 2009, 2012)). Recently there have been multiple attempts at quantifying verification complexity using graphical properties or subgraph patterns specific to verification (Jung & Salado 2023, 2024; Salado & Kannan 2019), both objective and subjective (Efatmaneshnik & Ryan 2016).

Verification Understandability. Cognitive reasoning is required whenever an exchange of information occurs. Engineers go through the same process with verifications of components or systems, engaging on different pieces of information to update their beliefs, judge the trustworthiness of the outcome, and then make decisions accordingly (Hamermesh 1985; Smith, Taylor & Sloan 2001). This manual information processing is prone to biases such as anchoring effects, causing rifts between the mathematically calculated and perceived belief scores (Tribus 1969). This research signifies the importance of subjectivity in the verification process, almost equal to the importance of the quantitative data collected throughout verification.

Different aspects of information processing are required for different types of system integration (Gold-Bernstein & Ruh 2004). Vertical integration offers a chain of information flow, where the received information tends to be functionally considered trustworthy. Horizontal integration utilizes communication interfaces to modularize involved subsystems, where data mismanagement such as excessive organizational information compartmentalization and misaligned interests can cause lower information trust. The system of systems paradigm leads to the lack of information flow between involved systems as they may have no direct relationships. Information processing strategies vary with system integration methods, resulting in different cognitive loads for engineers. It is assumed that such information processing strategy variations exist for the verification process as well, as it often runs parallel to the system development. Verification activities performed on individual system elements are used as the basis for the integrated system effectively being *integrated* themselves. System integration methods dictate the integration pattern for verification under this assumption, resulting in varying degrees of information processing necessities.

Previous work of the authors. We have conducted multiple research endeavors on verification complexity in scale, validating the representation and analysis of real-world industrial project verification strategies that are possible in graphical databases (Jung & Salado 2023, 2024). As there is no prior research on large-scale verification complexity measurements, the feasibility of the approach indicates the birth of new research directions in the field. Previous research was done to empirically evaluate the effectiveness of graph complexity measures used for verification complexity. The feasibility of graph-based empirical complexity measures was validated by detecting a number of distinctive graph complexity measures related to two industrial projects (Jung & Salado 2023). This was followed by an initial attempt at ordinal ranking of the aforementioned projects in terms of their verification complexities (Jung & Salado 2024). We have prepared a numerical ranking of multiple real and artificial verification strategies (Jung & Jung 2024).

The VCF and its Layers

VCF is proposed as a tool for measuring and managing verification complexity for industrial projects. The proposed VCF represents the complexity of verification as a three-dimensional block of complexity layers, various factors within a layer, hierarchy of layers, and evolution though time. It has been built by collating a plethora of systems engineering research on system complexity and verification processes. As there are few agreed-upon measures in verification complexities, we listed potential measures that could be used to represent complexity derived from each layer. We gathered measures in systems engineering and a wide range of relevant research domains from natural language processing to graph theories. These are suggestions at this stage for all but one layer; the eligibility and accuracy of these measures are to be evaluated in future research. We have existing research on the correlations between verification complexities and a set of graphical properties. The corresponding layer is therefore named as the *strategy complicatedness*, instead of strategy complexity, to clearly distinguish the partial focus these structural features represent.

The designs and implementation of verification strategies is a practical procedure with fluid execution considerations. The VCF is designed to satisfy such nature by allowing two characteristics: 1) the framework is designed at a high level to incorporate internal and environmental variations; 2) The framework can be nested, satisfying various necessities of nested and iterative verification following various system integration approaches.

The VCF visualizes the verification complexity layers as shown in Figure 1, including *foundation complexity, design challenge, ease of execution, strategy complicatedness, environmental,* and *information trustworthiness*. The horizontal dimension encompasses various complexity aspects within a given layer are omitted in the figure for better visualization. The distinction between *ease of execution* and *strategy complicatedness* is an example of complexity layers with parallel hierarchy, with both layers placed between the *design challenge* and *environmental* layers. Ther vertical dimension shows the layer hierarchy within the given level of encapsulation to which the VCF is applied. The middle layers with the grey background cover the complexity attributed to the verification being considered within the given VCF, while the bottom and top layers respectively point to the grounding factors dictating the underlying complexity derived from external factors and the complexity of communicating the verification results to others. The depth of the cube represents the evolution of verification complexity over time. This is loosely defined to incorporate various aspects of the time-related properties of verification complexities such as adaptation, familiarization, integration, training, progression, evolution, and so on. This malleability is designed to allow hierarchical stacking of the layers according to the structure of the system being verified.

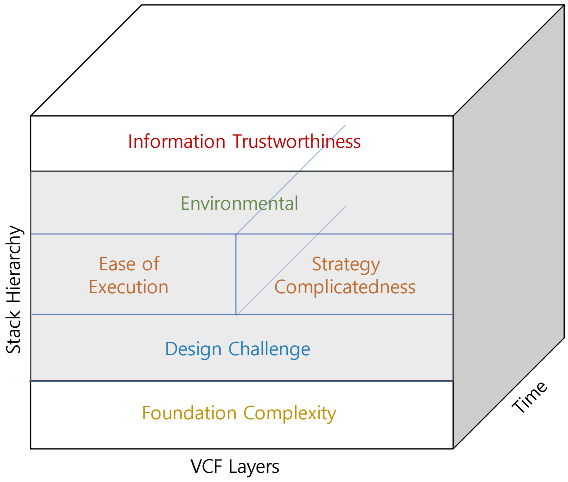

Figure 1. Overview of the Verification Complexity Framework layers.

The *foundation complexity* at the bottom is a base layer, providing an inlet for complexity factors generated outside of the verification process itself. This is mainly divided into two parts: systemic and intermodular. The systemic part represents the complexity of the system, where existing system complexity measures could be applicable. These include the system size, interconnectedness, orthogonality, predictability, and so on. The intermodular part reflects the verification complexity for other systems, which the currently verified system is dependent on (e.g. complexity of verifying a subsystem). The second part is added to incorporate the complexity of nesting multiple VCFs, as it is assumed that the verification complexity of a system is affected by the verification complexity of its components. Various human and environmental factors relevant to the system, such as training, sociopolitical barriers, and domain knowledge proficiency, are incorporated into system complexity. This is an innate complexity layer affecting all other layers and the complexity associated with them. The variety of practical considerations required in this layer indicates that automatic analysis may not be desirable. Measuring properties such as the proficiency of verification engineers and equipment performance quality requires system-specific manual analysis, for example, applying cognitive load theory for various human factors.

The top layer, the *information trustworthiness*, represents the difficulty in communicating verification results to external actors; this is to transfer the verification result information from one VCF to another, or to engineers. The overall complexity is first generated by aggregating the complexity for other layers of the VCF, which is then combined with the complexity of conducting a lossless information transfer to the receiver (be it the next VCF in the chain or the end user). The frequency and size of new information are one of the major properties of this layer as they are the main indicators of information transfer errors. Information transfer policies dictated by business logic for example play a major role in the degree of information mistreatment. Data explainability is added as a major property as it affects the trustworthiness of transferred information. Transparent reasoning behind the conclusion provides more context to the human users, improving their confidence in the received information. Types of connection (1-to-1, 1-to-n, n-to-n) are also considered in this layer as a factor. A range of mathematical and network methods could be applied to capture the complexity of outward information transfer (e.g., information entropy and Bayesian networks). The middle layers with the grey background divide the complexity of the verification into four layers. The *design challenge* layer covers the complexity of building the verification strategy, with the *ease of execution* and *strategy complicatedness* respectively representing complexities of executing individual verification activities and how they are interconnected. Lastly, the *environmental* layer contains relevant environmental factors.

The *design challenge* layer is proposed to encompass complexities generated from building the verification strategies. This includes the technical and personnel properties such as equipment availability, resource constraints, size of possible verifiable entities, selection of verification activities, documentation, and so on. Multiple qualitative properties are to be managed in this layer, for example through the Likert-scale surveys or the Cynefin framework.

The *ease of execution* and the *strategy complicatedness* layers deal with the complexity of implementing the designed verification strategies, each representing the complexity of verification activities and their interconnectedness respectively. This layer contains binary and numeric properties such as verification activity type and size, documentation accuracy, equipment and training availability, accessibility, data aggregation ratio, and so on. There is also a window of natural language processing with the semi-structured requirement matrix documentation. Lexical richness, for example, measures variations in document vocabularies between different systems (Tweedie & Baayen 1998). The *strategy complicatedness* layer embodies the complexity caused by how a verification strategy interconnects different verification activities. The size of verification, ratio of interconnected verification activities and requirements, verification orthogonality, and other graphical properties such as density, diameter, modularity, or connectivity are considered part of this layer. This is achieved by converting the verification strategy into a graph, with verification activities and relevant system as nodes with edges representing their connections. These two layers are placed side by side to signify their parallel characteristics; both represent the complexity of verification at the implementation stage.

The *environmental* layer is detached from the previous two layers as the factors considered within it have an overarching effect on the whole verification. This includes various external properties such as contractual, regulatory, lingual, and geopolitical factors. These are often qualitative in nature and therefore would benefit the most from manual measurements similar to that of the *design challenges* layer. The wider range of relevant factors indicates the survey could require more comprehensive audiences, incorporating system stakeholders on top of the engineers.

The VCF is designed to be stackable, capable of mimicking connection patterns derived from the various system integration paradigms. There are no limitations on how they can be nested together, therefore VCFs can form an identical graphical shape to the system being integrated. In this case, each VCF captures the verification complexity of individual system elements sharing the same topological position in the graph representation of the system integration. This is done by hierarchically feeding the outcome of the *information trustworthiness* layer as an input to the *foundation complexity* layer for another VCF as illustrated in Figure 2. These connections are not limited by their numbers, allowing 1-to-1, 1-to-n, and n-to-n connections between the VCFs. This unstructured graphical representability enables the VCF to be applied to all methods of system integration, up to and including the system of systems paradigm. The integrated system is verified as the cumulative outcome of interconnected VCFs, each with individual complexity levels.

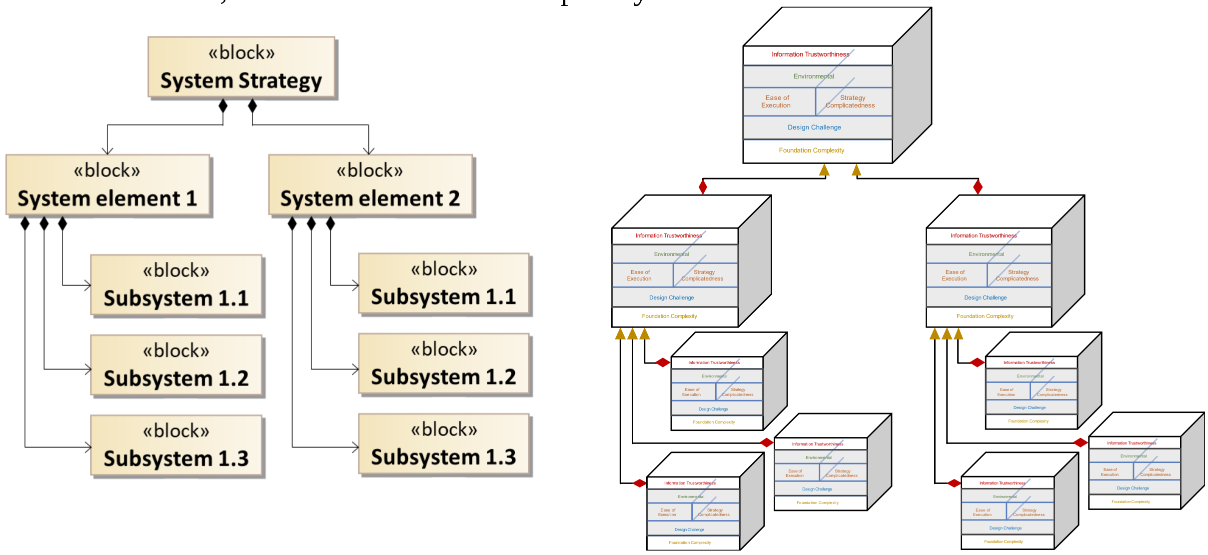

Figure 2. Illustration of an interconnected VCFs mimicking the product tree of an example system, linking verifications between system elements and subsystems.

Conclusion

This paper proposed the verification complexity framework as a means to formalize the definition of verification complexity in large-scale, real-world projects. Designed to be both static and dynamic, the framework offers nesting capabilities between them by transferring complexity measure between them. The VCFs have a built-in time axis to incorporate relevant verification evolutions throughout the system life cycle. The complexities represented by these layers are to be aggregated as an output of the given VCF. This value can be fed into another VCF as an input, contributing as a fundamental complexity surrounding the new VCF. Otherwise, it is transferred to the user as the final verification complexity value.

The proposed framework is conscious of the complexity of transferring information as well as the complexity of verification itself. The difficulty of lossless communication is incorporated into the framework to address the data explainability of the verification process pervasive to system development. The VCF can be used to provide transparent reasoning behind verification results, enhancing the trustworthiness of stakeholders throughout the decision making process. The semi-structured VCF design also provides adaptability to various information processing strategies employed by the verification engineers as well as the system users.

The VCF layers are in the empirical evaluation stage, where the ensemble-learning model implementation is prepared for machine learning experiments on quantitative measurements. Generation of a comprehensive series of artificial verification strategies is to be done to optimize the accuracy and coverage of the training result. The regression model is to be optimized with feature engineering once initial training is completed. Measures for other layers are to be formalized with additional literature review as well as expert opinion mining. We plan to focus on *design challenge* and the *environmental* layers as we are planning to introduce less resource-intensive, automatic measures that can represent these layers as well. The next step is the formalization of the complexity formula responsible for the generation of a singular complexity value for each VCF. Different impacts of complexity layers are to be evaluated by expert surveys and computational validations on existing datasets.

The VCF provides practical benefits by connecting the intangible qualities of system verification to quantifiable measures. This allows standardized complexity management in industrial environments. Understanding and organizing the complexity aids the planning and execution of verification strategies. Engineers would have access to a mathematical approach to verification complexity, leading to better understandability and communication. The VCF offers complexity management functionality to verification engineers. Verification structures can be optimized with mathematical comparisons, allowing holistic restructuring with minimal resources invested in each system evolution. Complex verification strategies can be tracked back to the complications in the systems leading to system design improvement as well.

The proposed framework likely has room for improvement, and we are planning on improving the VCF in the near future. Additional considerations are to be made for common underlying features such as time and human factors, enhancing the framework's generalizability. The time axis is to be formalized to represent the time and evolution within the VCF, and how changes in each layer affect others. The verification complexity measure would be refined in future research with the definition of zero complexity, allowing the ratio comparison between large industry projects in terms of their verification complexities.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CMMI-2205468.

References

Adcock, R, Sillitto, H & Sheard, S 2023, ‘Complexity’, *The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.9*, viewed <https://sebokwiki.org/wiki/Complexity>.

Alkan, B, Vera, D, Ahmad, M, Ahmad, B & Harrison, R 2016, ‘Design Evaluation of Automated Manufacturing Processes Based on Complexity of Control Logic’, *Procedia CIRP*, vol. 50, pp. 141–146.

Alkan, B, Vera, DA, Ahmad, M, Ahmad, B & Harrison, R 2018, ‘Complexity in manufacturing systems and its measures: a literature review’, *European J. of Industrial Engineering*, vol. 12, no. 1, p. 116.

Botchkarev, A & Finnigan, P 2015, *Complexity in the context of information systems project management. Organisational Project Management, 2 (1), 15–34*.

Brooks, SJ & Roy, R 2020, ‘A complexity framework for self-engineering systems’, *Smart and Sustainable Manufacturing Systems*, vol. 4, no. 3, ASTM International, viewed 8 March 2024, <https://openaccess.city.ac.uk/id/eprint/25288/>.

Buede, DM & Miller, WD 2016, *The Engineering Design of Systems: Models and Methods*, John Wiley & Sons.

Efatmaneshnik, M & Ryan, MJ 2016, ‘A general framework for measuring system complexity’, *Complexity*, vol. 21, no. S1, pp. 533–546.

Engel, A 2010, *Verification, Validation, and Testing of Engineered Systems*, John Wiley & Sons.

ESCC 2009, *ECSS-E-ST-10-02C – Verification, European Cooperation for Space Standardization*, viewed 14 March 2024, <https://ecss.nl/standard/ecss-e-st-10-02c-verification/>.

ESCC 2012, *ECSS-E-ST-10-03C – Testing, European Cooperation for Space Standardization*, viewed 14 March 2024, <https://ecss.nl/superseded-standard/ecss-e-st-10-03c-testing/>.

Fässberg, T, Harlin, U, Garmer, K, Gullander, P, Fasth, Å, Mattsson, S, Dencker, K, Davidsson, A & Stahre, J 2011, ‘An empirical study towards a definition of production complexity’, *International Conference on Production Research*, viewed 8 March 2024, <https://www.academia.edu/download/83007535/local_145754.pdf>.

Gold-Bernstein, B & Ruh, W 2004, *Enterprise integration: the essential guide to integration solutions*, Addison Wesley Longman Publishing Co., Inc., viewed 14 March 2024, <https://dl.acm.org/doi/abs/10.5555/993841>.

Hamermesh, DS 1985, ‘Expectations, life expectancy, and economic behavior’, *The Quarterly Journal of Economics*, vol. 100, no. 2, MIT Press, pp. 389–408.

Harel, D, Kupferman, O & Vardi, MY 2002, ‘On the Complexity of Verifying Concurrent Transition Systems’, *Information and Computation*, vol. 173, no. 2, pp. 143–161.

Horváth, B, Molnár, V, Graics, B, Hajdu, Á, Ráth, I, Horváth, Á, Karban, R, Trancho, G & Micskei, Z 2023, ‘Pragmatic verification and validation of industrial executable SysML models’, *Systems Engineering*, vol. 26, no. 6, pp. 693–714.

ISO 2023, ‘Systems and software engineering–system life cycle processes’, *ISO Standard*, vol. 15288–2023, viewed 14 March 2024, <https://www.services.bis.gov.in/tmp/WCLITD23321947_01062023_2.pdf>.

Jung, S & Jung, A 2024, ‘(Accepted) Verification Complexity: Definitions, Measurements, and Indicators’, *INCOSE International Conference on Human Systems Integration*.

Jung, S & Salado, A 2024, ‘Graph Complexity Measures as Indicators of Verification Complexity’, *Conference on Systems Engineering Research (CSER)*.

Jung, S & Salado, A 2023, ‘Verification Complexity: An Initial Look at Verification Artifacts’, *Conference on Systems Engineering Research (CSER)*, viewed 26 September 2023, <https://par.nsf.gov/biblio/10442450>.

Laing, C, David, P, Blanco, E & Dorel, X 2020, ‘Questioning integration of verification in model-based systems engineering: an industrial perspective⋆’, *Computers in Industry*, vol. 114, p. 103163.

Linehan, É & Clarke, S 2010, ‘Managing hardware verification complexity with aspect-oriented model-driven engineering’, *Proceedings of the 1st Workshop on Model Based Engineering* *for Embedded Systems Design (M-BED)*, Citeseer, pp. 54–60, viewed 13 March 2024, <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=fe0ce8c48ad5401136b55be7e73b098184addafd>.

Lyell, D & Coiera, E 2017, ‘Automation bias and verification complexity: a systematic review’, *Journal of the American Medical Informatics Association*, vol. 24, no. 2, pp. 423–431.

Rohloff, K & Lafortune, S 2005, ‘PSPACE-completeness of modular supervisory control problems’, *Discrete Event Dynamic Systems: Theory and Applications*, vol. 15, no. 2, pp. 145–167.

Romano, P 2009, ‘How can fluid dynamics help supply chain management?’, *International Journal of Production Economics*, vol. 118, no. 2, pp. 463–472.

Salado, A & Kannan, H 2019, ‘Elemental patterns of verification strategies’, *Systems Engineering*, vol. 22, no. 5, pp. 370–388.

Salado, A & Nilchiani, R 2014, ‘The Concept of Problem Complexity’, *Procedia Computer Science*, vol. 28, pp. 539–546.

Schoettl, F, Paefgen, M-C & Lindemann, U 2014, ‘Approach for Measuring Change-induced Complexity based on the Production Architecture.’, *Procedia CIRP*, vol. 17, pp. 172–177.

Sheard, SA & Mostashari, A 2013, ‘5.2.2 Complexity Measures to Predict System Development Project Outcomes’, *INCOSE International Symposium*, vol. 23, no. 1, pp. 170–183.

Sheard, SA & Mostashari, A 2011, ‘6.2.1 Complexity Types: From Science to Systems Engineering’, *INCOSE International Symposium*, vol. 21, no. 1, pp. 673–682.

Sinha, K, Suh, ES & de Weck, O 2018, ‘Integrative Complexity: An Alternative Measure for System Modularity’, *Journal of Mechanical Design*, vol. 140, no. 051101, viewed 18 March 2024, <https://doi.org/10.1115/1.4039119>.

Sinha, K & de Weck, OL 2016, ‘Empirical Validation of Structural Complexity Metric and Complexity Management for Engineering Systems’, *Systems Engineering*, vol. 19, no. 3, pp. 193–206.

Smith, VK, Taylor, DH & Sloan, FA 2001, ‘Longevity Expectations and Death: Can People Predict Their Own Demise?’, *American Economic Review*, vol. 91, no. 4, pp. 1126–1134.

Snowden, D 2010, ‘The cynefin framework’, *YouTube video*, vol. 8, p. 38.

Tribus, M 1969, *Rational Descriptions, Decisions and Designs Pergamon*, Elmsford, NY.

Tweedie, FJ & Baayen, RH 1998, ‘How Variable May a Constant be? Measures of Lexical Richness in Perspective’, *Computers and the Humanities*, vol. 32, no. 5, pp. 323–352.

Walden, DD 2015, ‘INCOSE Systems Engineering Handbook Version 4: A guide for system life cycle processes and activities’, *INCOSE International Symposium*, vol. 25, no. 1, viewed 14 March 2024, <https://cir.nii.ac.jp/crid/1130000795973147136>.

Weaver, W 1948, ‘Science and Complexity’, *American Scientist*, vol. 36, no. 4, Sigma Xi, The Scientific Research Society, pp. 536–544.

Wibben, DR & Furfaro, R 2015, ‘Model-Based Systems Engineering approach for the development of the science processing and operations center of the NASA OSIRIS-REx asteroid sample return mission’, *Acta Astronautica*, vol. 115, pp. 147–159.

Yin, X & Lafortune, S 2017, ‘Verification complexity of a class of observational properties for modular discrete events systems’, *Automatica*, vol. 83, pp. 199–205.

Zhang, Z 2012, ‘Manufacturing complexity and its measurement based on entropy models’, *The International Journal of Advanced Manufacturing Technology*, vol. 62, no. 9, pp. 867–873.