Verification Complexity: Definitions, Measurements, and Indicators

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 verification is an integral process of systems engineering, there is no consensus on measures for a full-scale verification complexity and what it represents. Verification engineers can rely on their implicit expertise to determine relative complexity differences, but this is resource intensive and scales badly with large systems. This research aims to define the verification complexity in an explicit and mathematical manner, suggest relevant measures, and propose indicators for verification complexity. Data gathering and experiment design have been finished with varying sizes and interconnections. Background research is being conducted on the verification complexity definition and relevant measures. Once finished, machine learning models will be trained on the measures with the proposed definition as a dependent variable. The trained model will then be analyzed to determine accurate, explicit indicators of verification complexity. These are expected to aid more accurate information propagation between system stakeholders, especially engineers, reducing system development costs.

Introduction

Verification utilizes various activities such as inspection, analysis, or testing to assess whether system requirements are successfully met (Engel 2010). Such assessment goes through a specific sequence, forming a verification strategy interconnecting requirements and their verification activities (Walden, Roedler & Forsberg 2015). As resource efficiency is one of the major considerations in system verification, reducing the complexity of such verification strategies is crucial. This requires engineers to accurately understand the degree of complexity, not only for the systems themselves but also for their verification strategies. We proposed that the verification complexity is dictated at least by the number of requirements to be verified, the number of possible verification activities that are planned, and their interconnections (Jung & Salado 2024a). Verification strategies were converted into knowledge graphs, and their structural properties analyzed for their correlation with their perceived complexity differences. With primary success with two real-world verification strategies, we are working on building a mathematical indicator for verification complexity. This is to be done by utilizing machine learning techniques, training a model for the verification complexity using various verification complexity measures as input. Graph complexity measures were shown to be possible candidates for such measures as verification strategies can be viewed as knowledge graphs transferring information between nodes (Jung & Salado 2023a).

There are two tracks to this research: 1) implementation and training of the machine learning model for verification complexity indicators, and 2) defining the verification complexity itself as well as determining relevant measures. The first research track is nearing the end of the preparatory stage, gathering data and technology for experimentation. There exists an initial experiment plan to learn a mathematical verification complexity indicator with manually stated complexity orders between the different verification strategies; we are working on getting expert opinions on the matter. Verification complexity in this case is defined as an ordinal measure, manually interpreted by analyzing verification strategies. This is not a scalable approach, however, as manual interpretations are often inconsistent.

We are working on proposing a less implicit definition of verification complexity as a prerequisite to the main experiment. The idea is to define verification complexity philosophically and mathematically in a tangible format with data-driven measurements. A wide range of systems engineering domains will be addressed, from the structure of verification strategies to human factors related to the verification engineers. A list of measures will be deemed relevant to the verification complexity based on this definition, and a machine learning model will be trained to find combinations that best represent the defined verification complexity. This will reduce, if not remove, inconsistent measures while improving their accuracy. The measuring guidelines will be proposed to allow verification engineers to reproduce the same complexity values in identical verification processes.

The machine-learning technologies and verification strategy graphs gathered in the initial stage of the research can then be re-used to train a verification complexity indicator. We expect this to be a low-cost, fast, and reliable complexity indicator aiding verification engineers by providing a mathematical medium for determining and communicating complexity. This would complement the manual verification complexity analysis, which requires a great deal of time and resources at the cost of accuracy.

This paper is organized as follows. The Verification Strategy Graphs section discusses the dataset, verification strategy graphs gathered from ten existing literature as well as a description of the two real-world verification strategies from industry projects. The Machine Learning Models section describes the use of ensemble learning models and the reasons for it. This research is still a work in progress and there is no Result section validating the effects of the proposed approach. The Conclusion and Remaining Works section provides a summary of the current research progression as well as the planned future works.

Verification Strategy Graphs

Graph complexity measures have been used as a measure of system complexity and are expected to be likely candidates for verification complexity measures as well. To calculate these measures, the verification strategies extracted for experiments are converted into graphs. We utilized ten verification strategies previously found in existing literature (Jung & Salado 2023b). This verification strategy design literature utilized artificially simplified *toy problems* with smaller problems and fewer interdependencies between the entities. Literature was reviewed to capture the number of distinct requirements and verification activities in their strategies as well as the interconnections within them. Since the graphical structure was integral to this research, explicit entity and relation definitions were used to gather information. Table 1 shows the ten graphs made out of these ten data sources. The source articles were searched for predefined graphical formats of their verification strategies; if the article did not have such graphs, the manuscripts were read for specific textual definitions. Graph1 and Graph6 both resulted in edgeless graphs, as neither source specified the relationships between V&V activities and their requirements. Graph4 and Graph5 resulted in null states, as their strategies required additional interpretations; this included distinguishing requirements over different development stages, connecting requirements with verification evidence, and so on. Graph measures cannot be calculated on empty or edgeless graphs; therefore, the remaining six verification strategy graphs were selected for experimentation.

The size of these six graphs ranges from six nodes with seven edges to 51 nodes with 72 edges. These *toy project* graphs are combined with two larger verification strategies from actual industrial settings. These include verification activities for a non-pharmacological physical treatment device (*pharma*, with 404 nodes and 563 edges) and for a defense system product (*defense*, 8,922 nodes and 17,319 edges). These graphs are expected to represent various complexity difference patterns, such as significant size differences (between graphs from existing literature and real-world systems) and significantly different structures between similar-sized verification strategies (within graphs from existing literature). We utilized the graph structures because its scalability has been shown with these datasets (Jung & Salado 2024b); unlike the manual approach, mathematical approaches can be applied to real world projects regardless of their requirement sizes. This is especially relevant as we believe it is likely that verification engineers might experience cognitive overloads when trying to reason through such large networks of data for verifying such large systems.

Table 1: List of Verification Strategies from Existing Literature with Graphical Size Attributes.

Graph	Source	# Nodes	# Edges
Graph1	(Barad & Engel 2006)	11	0
Graph2	(Salado & Kannan 2019)	51	72
Graph3	(Salado, Kannan & Farkhondehmaal 2019)	10	15
Graph4	(Kulkarni et al. 2020)	-	-
Graph5	(Kulkarni, Wernz & Salado 2021)	-	-
Graph6	(Kulkarni et al. 2021)	7	0
Graph7	(Xu & Salado 2019)	6	7
Graph8	(Xu, Salado & Xie 2020)	14	17
Graph9	(Xu, Salado & Deng 2022)	49	69
Graph10	(Xu & Salado 2023)	12	11

Machine Learning Models

There lacks a full-scale verification complexity analysis in the field to our knowledge (Jung & Salado 2023a). We are pioneering the definition of scalable and explicit verification complexity. As such, the exact format of such a definition is malleable at the moment. Verification complexity could be defined as an accurate ratio model representing intervals between different verification strategies with a definition for zero-complexity strategies. On the other hand, it could be limited to an ordinal model only capable of distinguishing relative differences between them. Measurements would experience similar issues as both quantitative and qualitative measures can be designated relevant. As such, we are using an ensemble learning model compatible with various dependent variable formats.

The ensemble method combines various machine learning algorithms such as gradient boosting, random forest, boosted classifier, and so on. These algorithms can be selectively utilized to best match the set of dependent and independent variable formats. The selected algorithms are then randomly combined and run independently to detect an optimal combined training result, which is then used to train metalearners stacked on the higher hierarchy (Laan, Polley & Hubbard 2007). The outcome is a collection and distribution of selected training algorithms for the given training task. Currently, Python's scikit-learn[1] and H2O AutoML[2] libraries are used to implement the machine learning models. Both libraries are comprehensive enough to grant basic data preprocessing as well as iteratively training the models with intermittent model tunings. Since we are planning to have a clearly defined dependent variable (verification complexity), lower emphasis has been given to the feature engineering task. Simpler regression algorithms such as linear and polynomial regressions are considered major candidates for their high interpretability. This is to provide more evidence to the estimated verification complexity, increasing the trustworthiness of the resulting measure.

[1] *https://scikit-learn.org/stable/*
[2] *https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html*

Conclusion and Remaining Works

Training on the manually attributed verification complexity is nearly done with the expert analysis of the model remaining. Necessary experiment preparations are finished including data preprocessing and program implementation. This part of the research is expected to be done in the near future with no significant resource investment.

We are conducting multiple research relevant to this paper. Proposing the definition and relevant measure for the verification complexity is still in its background research stage, with the proposal of Verification Complexity Framework (Jung & Salado 2024a). We are also researching cognitive psychology to discuss information transaction iterations occurring between the verification engineers (Jung & Salado 2024c). These transactions naturally influence the information in future transactions between verification activities and requirements (Salado & Kannan 2018). More frequent processing results in higher cognitive loads, increasing human errors especially in large systems with complex problems (Dörner & Güss 2022; Kirsh 2000). We aim to incorporate cognitive aspects to improve the verification complexity measures, addressing cognition bias errors caused by cognitive overloads (Jung & Salado 2024b).

Acknowledgements

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

References

Barad, M & Engel, A 2006, 'Optimizing VVT strategies: a decomposition approach', *Journal of the Operational Research Society*, vol. 57, no. 8, Taylor & Francis, pp. 965–974.

Dörner, D & Güss, CD 2022, 'Human error in complex problem solving and dynamic decision making: A taxonomy of 24 errors and a theory', *Computers in Human Behavior Reports*, vol. 7, p. 100222.

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

Jung, S & Salado, A 2024a, '(Accepted) Exploring A Verification Complexity Framework', *INCOSE International Conference on Human Systems Integration*.

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

Jung, S & Salado, A 2023a, *(Submitted) Emergent Knowledge Patterns in Verification Artifacts*.

Jung, S & Salado, A 2024c, '(Submitted) Improving Verification Strategies for Cognitive Load Management', *INCOSE International Conference on Human Systems Integration*.

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

Kirsh, D 2000, 'A Few Thoughts on Cognitive Overload', *Intellectica*, vol. 1, no. 30, pp. 19–51.

Kulkarni, AU, Salado, A, Wernz, C & Xu, P 2020, *Is Verifying Frequently an Optimal Strategy? A Belief-Based Model of Verification*, American Society of Mechanical Engineers Digital Collection, viewed 16 January 2024, <https://dx.doi.org/10.1115/DETC2020-22582>.

Kulkarni, AU, Salado, A, Xu, P & Wernz, C 2021, 'An evaluation of the optimality of frequent verification for vertically integrated systems', *Systems Engineering*, vol. 24, no. 1, pp. 17–33.

Kulkarni, AU, Wernz, C & Salado, A 2021, 'Coordination of verification activities with incentives: a two-firm model', *Research in Engineering Design*, vol. 32, no. 1, pp. 31–47.

Laan, MJ van der, Polley, EC & Hubbard, AE 2007, 'Super Learner', *Statistical Applications in Genetics and Molecular Biology*, vol. 6, no. 1, De Gruyter, viewed 22 January 2024, <https://www.degruyter.com/document/doi/10.2202/1544-6115.1309/html>.

Salado, A & Kannan, H 2018, ‘A mathematical model of verification strategies’, *Systems Engineering*, vol. 21, no. 6, pp. 593–608.

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

Salado, A, Kannan, H & Farkhondehmaal, F 2019, ‘Capturing the Information Dependencies of Verification Activities with Bayesian Networks’, in S Adams, PA Beling, JH Lambert, WT Scherer & CH Fleming (eds), *Systems Engineering in Context*, Springer International Publishing, Cham, pp. 581–591.

Walden, DD, Roedler, GJ & Forsberg, K 2015, ‘INCOSE Systems Engineering Handbook Version 4: Updating the Reference for Practitioners’, *INCOSE International Symposium*, vol. 25, no. 1, pp. 678–686.

Xu, P & Salado, A 2019, ‘A Concept for Set-based Design of Verification Strategies’, *INCOSE International Symposium*, viewed 16 January 2024, <https://onlinelibrary.wiley.com/doi/abs/10.1002/inst.12425>.

Xu, P & Salado, A 2023, ‘A Mathematical Approach to Design Verification Strategies That Incorporate Corrective Activities as Dedicated Decisions’, *IEEE Open Journal of Systems Engineering*, vol. 1, pp. 1–11.

Xu, P, Salado, A & Deng, X 2022, ‘A Parallel Tempering Approach for Efficient Exploration of the Verification Tradespace in Engineered Systems’, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 11, pp. 7223–7235.

Xu, P, Salado, A & Xie, G 2020, ‘A Reinforcement Learning Approach to Design Verification Strategies of Engineered Systems’, *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3543–3550, viewed 16 January 2024, <https://ieeexplore.ieee.org/abstract/document/9282929>.