A Procedure for Exploring Detrimental Weak Emergent Behavior in Complex Systems of Systems

Rune Andre Haugen
Dept. of Science and Industry Systems
University of South-Eastern Norway
Kongsberg, Norway
rune.a.haugen@usn.no

Gerrit Muller
Dept. of Science and Industry Systems
University of South-Eastern Norway
Kongsberg, Norway
gerrit.muller@usn.no

Satyanarayana Kokkula
Dept. of Science and Industry Systems
University of South-Eastern Norway
Kongsberg, Norway
satyanarayana.kokkula@usn.no

This is a pre-copyedited version of a contribution published in 2024 19th Annual System of Systems Engineering Conference (SoSE) published by IEEE. The definitive authenticated version is available online at IEEE Explore: 10.1109/SOSE62659.2024.10620939

***Abstract*—Industrial companies developing complex systems have face challenges with undesired unforeseen system behavior emerging in late development stages or during the system's operational use. This paper proposes a systematic approach from a Systems Engineering perspective to overcome these challenges. Combining Design of Experiments with regression analysis while conveying a beneficial human vs machine task balance enables us to shift the focus from individual requirements to the overall system design for system testing without overwhelming efforts. We aim to keep the specific performance stated through requirements while ensuring a minimum performance throughout the parameter space. Actively using measurements during the development enables monitoring of the system performance throughout the parameter space facilitating detection and subsequent elimination or reduction of inherent detrimental emergent behavior. The proposed procedure also gives a solid rationale for the case company in what to test and not.**

Keywords—Automation, Complex Systems, Design of Experiments, Emergent Behavior, Systems Integration, Systems of Systems

I. INTRODUCTION

A. Background

Case company: The KONGSBERG group [1] spans multiple businesses with its 13 000 employees. The 210 year old company has multiple global sites headquartered from Kongsberg. We did our research in one of the KONGSBERG subsidiaries henceforth referred to as the case company.

Products: The context of this paper is generic complex cyber-physical Systems of Systems (SoS), based on real case company products in their operational environment, as the real cases we investigated contain sensitive data that cannot be shared outside the case company. The real product portfolio spans a wide range from deep sea to outer space. Products developed for the defense and maritime sectors are typically part of a larger network of complex ecosystems.

Problems: The parameter space of this class of systems is infinitely large and exhibits a tremendous amount of emerging behavior, most of which is considered unforeseen by systems designers. Capturing unforeseen undesired emerging behavior through testing and analysis takes (too) many resources yet contributes to reducing risk of unacceptable operational failure.

System performance is often characterized by a large parameter set that together form a parameter space. The product specification can describe the entire n-dimensional parameter space, or it can describe only a limited number of relevant points. Typically, we define each requirement as being independent of other requirements. In reality, ignoring requirements dependencies results in underspecified systems. Design and testing to cover the superset makes the system either too expensive or impossible. In other words, users know and expect that the systems have limitations in their performance. Developers can help themselves by limiting the system rather than opting for a system without limitations. Fully specifying all use cases and limitations obfuscates the specification. Therefore, we are balancing a level of specification detail that we can understand and that will result in acceptable results. The problem is that in the intermediate acquisition, tendering, and acceptance process, the contractual black and white statements replace the beforementioned nuance. Trust and interaction between stakeholders is essential to ensure that we develop the right system. However, the formal rules (e.g., the above processes) and difficulties in changing company culture are increasingly in the way of doing the right things.

The current practice in the case company is that their test process results in verified products but offers limited early validation. The most thoroughly tested products within the case company are space systems because these systems are extremely expensive to repair in operational use. However, many products from the company have been in operational use for a long time, exhibiting high performance, and are therefore validated by users.

Needs: The case company needs to improve the way of testing during Systems Integration to increase the detection of significant undesired non-intuitive emerging behavior with an order of magnitude, without spending more resources. Further, they need to map the system performance throughout the parameter space to gain sufficient situational awareness to

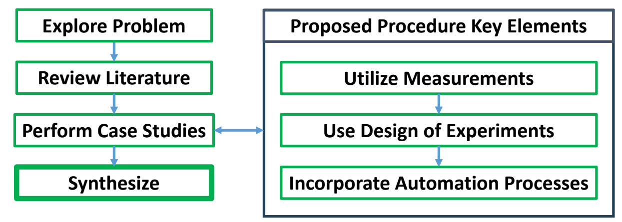

Fig. 1. Simplified research design

This research was funded by the Research Council of Norway under grant number 321830.

facilitate proper mitigation efforts. This will reduce the gap between the case company practice and the state-of-art regarding early validation of products and also improve product robustness.

B. Research Design and Questions

Fig. 1 shows a simplified research design. The research in this paper builds on previous research through two problem space exploration studies [2, 3], a literature review [4], and four case studies [5-8]. We aim to answer the following research questions in this paper:

- **Research Question (RQ):** How does the company get better at robustly achieving the defined measurements?
- **Sub-Research Question 1 (SRQ1):** What is the deviation between the current company practice and the state-of-art?
- **Sub-Research Question 2 (SRQ2):** What should the company do to move closer to the state-of-art?

C. Research Positioning and Framing

We conducted this research within the field of Systems Engineering, and more precisely the sub-field of Systems Integration. We used a synthesis approach to establish a feasible procedure for the case company based on our previous research, supplemented by the industry-as-laboratory [9, 10] approach to validate the feasibility of our proposed procedure through discussions with Subject Matter Experts (SMEs) in the case company. This company develops complex systems for a wide range of users and use cases. We can characterize these products as cyber-physical systems. The systems interact with both human operators and the environment in which they operate, expanding the frame to a SoS level. See Fig. 2 for a specific case where an Autonomous Surface Vessel (ASV) operates at sea to detect a vessel in distress. This case involves one complicated physical system (ASV) and two complex systems (the physical vessel in distress with human operators and the operational environment), together forming a complex cyber-physical SoS.

We root our proposed procedure in the Systems Engineering Vee model and measurements, which are described in common Systems Engineering literature like the *Systems Engineering Handbook* [11] and the *Systems Engineering Body of Knowledge* [12]. Measures of Effectiveness (MoEs) are the customer's key indicators of achieving mission success, Measures of Performance (MoPs) measure attributes important to ensure operational objectives as performance necessary to meet MoEs, and Technical Performance Measures (TPMs) measure critical technical parameters derived from the MoPs [11].

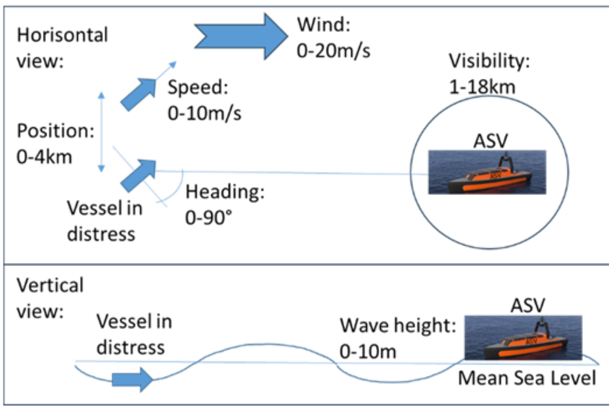

Fig. 2. Research case

The proposed procedure utilizes Design of Experiments (DoE) in tandem with regression. DoE is a technique to define and investigate selected test points within all possible combinations in an experiment involving multiple parameters [13, 14], while regression in our case is to find the effects of input parameters on an output parameter [14].

D. Main contribution and scope of research

The main contribution of this paper is to synthesize a procedure for exploring detrimental weak emergent behavior in complex SoS, building on innovations from our previous work [2-8].

The remainder of this paper is structured as follows: The section on Synthesis proposes a procedure for exploring detrimental weak emergent behavior in complex SoSs based on obtained knowledge from our previous studies [2-8]. The section Discussion provides answers to the research questions in light of the synthesis. Finally, the section Conclusion summarizes the paper, provides gained knowledge, defines limitations of the study, and proposes future research.

II. Synthesis

A. Problem Space Exploration

We base the problem space exploration on two papers researching the case company's major challenges [2, 3].

As discussed in Kjeldaas, et al. [2], the case company experiences challenges with detection of emergent behavior during system testing. System behaviors that emerge from the interaction of sub-systems can be difficult to capture through system requirements testing. High cost of performing tests involving repetitive manual processes handled by already overloaded SMEs causes the company to focus on requirement testing. The result of this practice is potential late error detection and subsequent project delays. Kjeldaas, et al. [2] suggest broadening the focus from requirements testing, automating current manual tasks that are causing bottlenecks in the test process, and improve communication and scheduling between disciplines to facilitate better cooperation and understanding.

Skreddernes, et al. [3] discuss the challenges with verification in complex engineered product development. Complex product development involves a high degree of uncertainty that manifests in rapid iterations, which challenges the interdisciplinary project organization. As the case company focuses on rapid improvement cycles in the system design, this poses challenges for them keeping up with regression testing. The result of this practice is a higher risk in the integration phase that could impose undesired uncertainties also in verification testing. Skreddernes, et al. [3] suggest scheduling for sufficient integration- and regression test activities and using measurements to monitor the system performance.

B. Supporting Literature

A paper researching the literature related to emergent behavior and its detection forms the basis of this section [4].

The literature describes emergent behavior as behavior at macro level resulting from interactions among the constituent parts at micro level that cannot directly be traced back to the micro level. Mittal, et al. [15] categorize emergent behavior along a scale from simple, through weak and strong, to spooky. The emergent behavior categories correspond to complexity categories ranging from simple, through complicated and complex, to chaotic. Weak emergent behavior is predictable, but non-intuitive. The difference between complicated and complex systems is that complicated systems are repeatable while complex systems are not. We typically see weak emergent behavior in complicated and higher complexity systems. One other important aspect is that observers can see emergent behavior at different levels of this scale depending on their experience. Kopetz, et al. [16] bring another aspect to emergent behavior as being beneficial or detrimental in combination with being expected or unexpected.

There are many methods proposed in various sources to deal with emergent behavior, where Haugen, et al. [4] indicates three main directions:

Test coverage involves using methods like DoE and temporal logic to extract maximum amount of information from minimum amount of tests.

Manual and automatic tasks involve exploiting both human and machine strengths. Humans are good at reasoning while machines are more efficient in performing routine operations. Diallo, et al. [17] proposes statistical debugging, while Raz, et al. [18] and Raman and Jeppu [19] propose Machine Learning as method to extract emergent behavior.

Combined modeling and simulation efforts typically involve Agent-Based Simulations [20, 21] to detect emergent behavior and subsequent model tracing to provide insight into the causes of the observed emergent behavior [22, 23].

C. *Case Studies*

We base this section on four case studies in the company [5-8], where [7, 8] are “under review” soon to be published.

Haugen and Mansouri [5] used a systems thinking approach to explore how automation processes would affect the test process and the organization of the case company, which indicated a potential for increasing the efficiency up to fifteen times compared to its current test process. A positive consequence of this increase in efficiency is that the employees can spend more time on other pressing tasks.

Haugen and Ghaderi [6] used a DoE approach to detect expected detrimental emergent behavior, which showed to be 5-50 times more effective compared to the current test strategy in the case company. This higher detection rate of undesired behaviors will provide the company with more data that can help increase their understanding of these negative system behaviors.

Haugen, et al. [7] researched how automation processes had taken effect in a case company project, which proved to increase the efficiency of test result analysis in an ongoing development project by approximately 30%. Test setup, test execution, and test result documentation showed to be beneficial to automate in order to increase the efficiency of the company’s test process in addition to test result analysis.

Haugen, et al. [8] explored unexpected detrimental emergent behavior of a complex system considering noise parameters. DoE in tandem with regression analysis both detected and mapped detrimental weak emergent behavior to increase the system knowledge of the development project in the case company. This approach enables a fast track to a coarse situational awareness of the system’s performance throughout the parameter space, and the possibility to obtain a higher fidelity in this coarse scan overview.

D. *Proposed procedure*

This section on proposed procedure is building on the research from the previous sections. Fig.3 illustrates the proposed procedure connected to the Systems Engineering Vee model through measurements and feedback. We have developed the proposed procedure mainly for the system integration phase and traced the steps to our previous work (references above the steps/boxes in Fig.3).

We start in the Vee model with step 1-3 in Fig.3. Obtaining and understanding stakeholder needs aids in determining measurements. These measurements do not need to be as strict

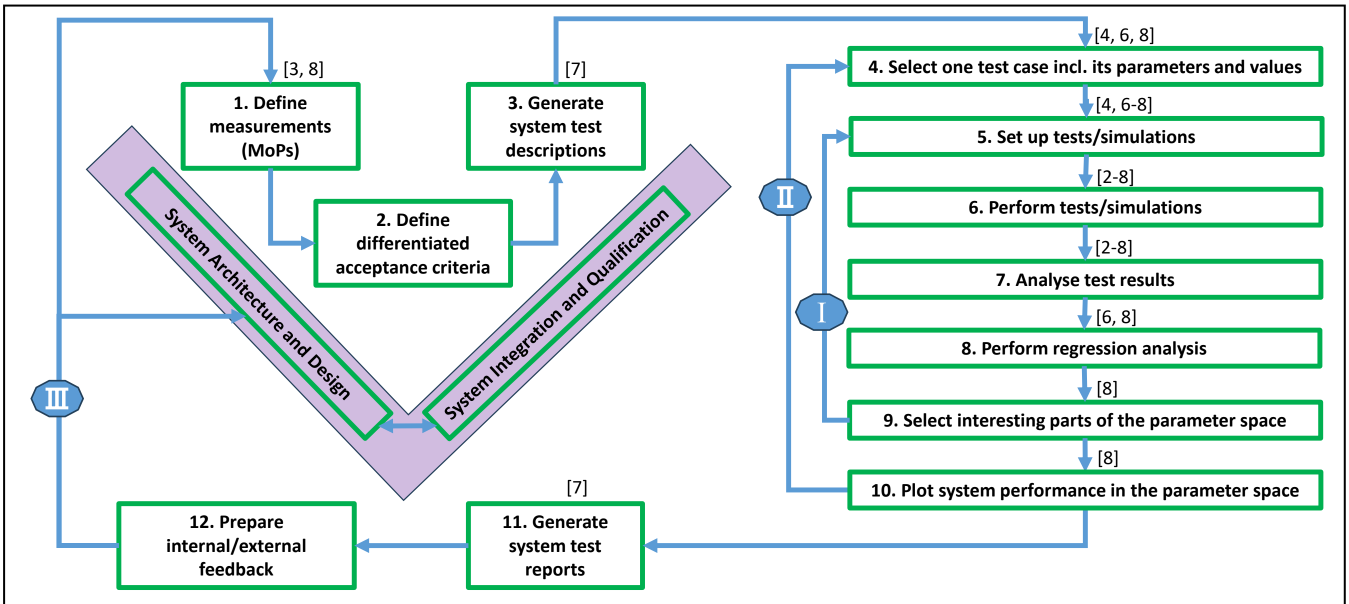

Fig. 3. Proposed procedure

as the constrained system requirements, as the acceptance criteria can be different with and without specific constraints. To provide input to testing, the project should make test descriptions. Machines generate test descriptions more efficient than humans, which the case company can facilitate semi-automatically through their system owners team (architects, designers, and integrators) and Highly Automated Document System (HADES) [7].

We continue with the test procedure in steps 4 through 10 in Fig.3. The system development team (the system integrator team supported by the system design team) manually select one test case, define its relevant independent parameters, and define the parameters' corresponding minimum and maximum values [8]. Higher knowledge within the system owners team will help them faster get to the desired level of information with as few tests as possible. However, omitting significant parameters can render the testing useless at worst. Guided by a super-user, the system owners team can set up experiments semi-automatically using the Minitab tool [24], using Orthogonal Arrays (OAs) through the Taguchi method or two-level Fractional Factorial DoE [7, 8]. Then, transfer the test setup to the test arena (simulator). A System Integrator plans the use of test arenas and test personnel. Testers can run simulations either manually or automatically, depending on the capabilities of the test arena, based on the provided test setup. After a test that is in need for analysis, the tester stores the test outcome in a test results database. The system owners team obtains a specific set of test result analyses whenever deemed necessary. The system owners team can conduct test result analyses manually or preferably HADES can do so automatically and store the results in a specified server location [7, 8]. See Fig. 4 for an overview of the case company's current test process [8]. Guided by a super-user, the system owners team can transfer the test result analysis results to the Minitab tool and perform a regression analysis to extract the parameter effects [8]. The effects found through this semi-automatic process can guide the system owners team further in their search for detrimental weak emergent behavior. Based on the obtained information of parameter effects, the system owners team can select interesting parts of the parameter space for further investigation. We base this new iteration, seen as **iteration Ⅰ** in Fig. 3, on the following information [8]:

- the parameters that have a significant effect
- the effect that part of the parameters' value range has

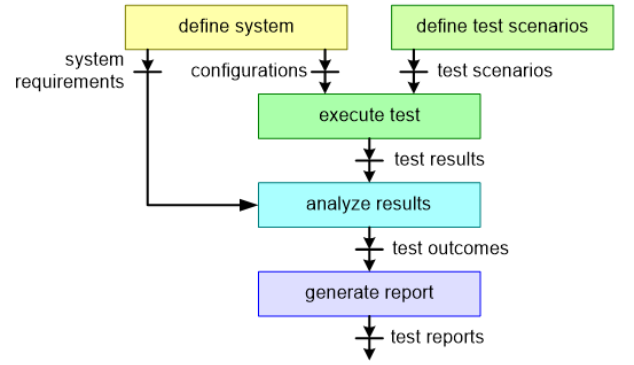

Fig. 4. Current company test process

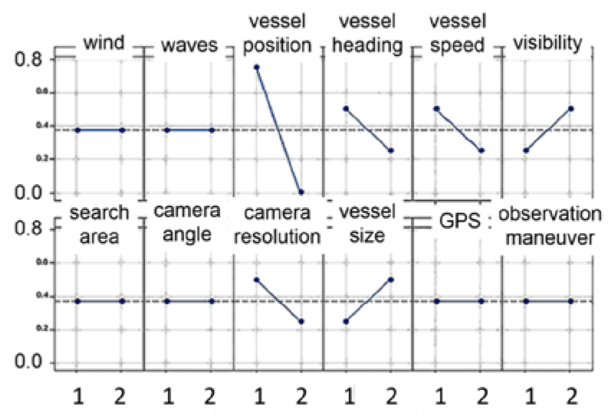

Fig. 5. Parameter effects

- the parameter value combinations that give a response below the defined performance threshold

See Fig. 5 for an example where we see what parameters show what magnitude of effect from a L16 Taguchi OA based set of simulations, where six out of twelve parameters show an effect [8]. The system owners team together with the Systems Integrator coordinate the desired amount of iterations to obtain the necessary level of information [8]. They aim to have a good situational awareness of the system performance related to a specific test case throughout the parameter space of interest. To achieve this situational awareness, the system owners team can perform regression and plotting semi-automatically in the Minitab tool to map the system performance level [8]. The knowledge of super-users might be necessary to ensure a good practice for plotting the results and interpreting them appropriately according to related uncertainties. The system owners team coordinates the above steps for all relevant test cases of the system in close cooperation with the System Integrator, seen as **iteration Ⅱ** in Fig. 3.

We finish in the Vee model with step 11-12. The system owners team can use HADES to create test reports semi-automatically to provide test results on a human readable format for both internal and external usage [7]. The System Architect and System Integrator facilitate a feedback loop to stakeholders in the development project, seen as **iteration Ⅲ** in Fig. 3, typically to the system design group, for conveying the rationale for potential efforts to mitigate any detrimental weak emergent behavior. It may also be necessary to involve customers to discuss adjustments to the defined measurements and acceptance criteria. The System Architect and System Integrator together with other management roles (Project Manager, Test Manager, and Chief Engineer) manage iterations to ensure that test results comply with defined acceptance criteria.

The system owners team together with the System Integrator oversee the final round of testing as a formal verification of the system in the qualification phase where we bypass step 8 through 10 in Fig.3.

See Fig. 6 for an illustration of how we can use this approach. For the considered SoS with a defined MoP (4km vessel position offset), we see how the ASV's detection window (MoE) changes as a function of the MoP. The black (solid) region represents the parameter space we have tested,

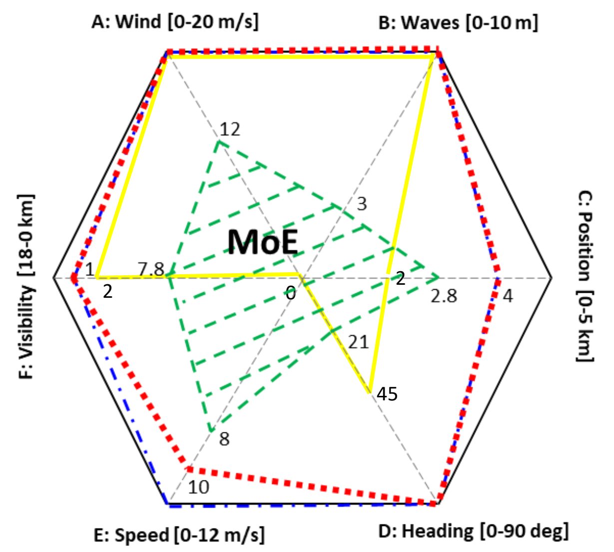

Fig. 6. System performance overview

the red (dotted) region represents the parameter space the ASV is designed to operate, the blue (dash-dot-dash) region represents the parameter space the ASV is conditionally (dependent on parameter value combinations) achieving the MoE, and the green (dashed/hatched) region represents the parameter space the ASV is always performing according to the MoE. The yellow region indicates one extreme case where we look at what the ASV can do in a harsh environment (strong winds, high waves, and low visibility). If we have a situation with 2km visibility and the vessel in distress has a heading offset of 45degrees, we see that the ASV will not meet the MoE when we set the vessel position offset MoP to 4km. If we can reduce that MoP to 2km, we can validate this ASV performance for the yellow region. The measurements will help the development team to keep focus on the customer expectations. As part of defining system requirements, the project should also define the related test cases and their acceptance criteria [11]. The System Architect and System Integrator roles working across "silos" can help this process succeed. These two roles should work closely together to iteratively define requirements and corresponding test cases. The definition of test cases should further cover the MoEs and corresponding MoPs throughout the parameter space. It is when we look at the superset of requirements, not isolated requirements, that we typically experience detrimental weak emergent behavior [8].

We see a generalization of the proposed procedure in Fig. 3, establishing a common procedure to explore detrimental weak emergent behavior in complex SoS. Even though the *Systems Engineering Handbook* [11] describes measurements, we believe our usage of these in the proposed procedure is novel.

III. Discussion

Integration should start with identifying measurements. The essence of integration is to increase the probability of achieving the MoEs and MoPs as early as possible. SMEs should therefore determine the main risks for achieving them and transform that into tests / experiments that help detect undesired behavior early and vice versa increase confidence early. In essence, this is a risk-based approach.

Using DoE with OAs [14, 25] can exploit tests an order of magnitude better regarding effectiveness than current practice at the case company [8]. Additionally, automation processes [26, 27] could bring another order of magnitude improvement regarding efficiency. The proposed procedure is a multi-phase approach, e.g., a coarse scan to find problems followed by zooming in on these to investigate the problem areas. The SMEs should determine for these tests what "coarse" means, e.g., what parameter ranges and step-sizes initially and when zooming. These phases allow an iterative way of testing to first screen for detection of detrimental (undesired) weak (unforeseen) emergent behavior and then investigate for mapping it in the parameter space. The proposed procedure provides us with a diagnostic tool to diagnose and enable treatment of the undesired system behavior based on information of problem areas' location, size, and shape in the parameter space.

To achieve desired performance throughout the parameter space set for the system to operate, we propose a procedure for exploring detrimental weak emergent behavior in complex SoS. Compared to current practice for the case company, this will involve broadening the horizon from merely focusing on constrained system requirements to also focusing on system performance throughout the parameter space. Setting different constraints for requirements and generally for the parameter space can ensure specific performance for parts of the parameter space where this is possible and feasible.

Furthermore, automation processes can reduce current bottlenecks (test setup, test execution, test result analysis, and test related documentation) in the company's test process and thereby leverage their capability to perform integration testing and regression testing. A procedure using OAs to cover the parameter space further secures the case company a solid rationale for their testing, which they can use both as internal guidance to schedule sufficient testing and to convince their customers that the test effort ensures good test coverage.

We re-visit the research questions to provide answers, starting with the sub-research questions.

SRQ1: What is the deviation between the current company practice and the state-of-art?

There is a delta between the current case company practice and the state-of-the-art. The company is following the Systems Engineering process but focusing on system requirements constrained to mostly "sunny day" and "corner case" conditions instead of system performance covering the problematic areas of the parameter space using MoPs. The case company experiences challenges related to integration- and regression testing because of increasing product complexity and higher workload through more projects in parallel, calling for an effort to reduce the gap to the state-of-art.

SRQ2: What should the company do to move closer to the state-of-art?

The company should increase both the effectiveness and efficiency of their test process to ensure more robust products and cope with the foreseen future workload. Regarding effectiveness, the company should utilize the power of DoE and regression to extract maximum information in minimum amount of tests. The proposed approach will cover both traditional requirements testing as well as the overall parameter space testing. Regarding efficiency, the company should automate tasks where machines outperform humans. This automation approach will also facilitate a higher resolution in the DoE approach compared to more time-consuming manual operations, as to run more tests during a defined period.

RQ: How does the company get better at robustly achieving the defined measurements?

The case company verifies TPMs related to MoPs for parts of the parameter space, but they need to cover more of the problematic parts of the parameter space. The proposed procedure will help them achieve this by increasing their confidence upwards from TPMs, through MoPs, to MoEs for interesting parts of the parameter space. This coverage may involve multiple levels of the different MoPs for various regions of the parameter space. In some regions it may even be physically impossible to achieve the MoEs, both intuitive and non-intuitive, which is crucial information for both the company and the users.

IV. Conclusion

Industrial companies engineering complex systems have a big challenge with inherent detrimental weak emergent behavior, being non-intuitive undesired system behavior manifesting in late project stages or operational use of the product. The procedure proposed in this paper will leverage the effectiveness and efficiency of the case company's test process and ensure early detection of undesired system behavior and thereby reduce project risk. DoE and regression provide a toolset to effectively detect problem areas, and automation processes will help doing so faster and/or with less uncertainty regarding undetected problems. Through a sensitivity analysis of the parameter space related to defined MoPs, we can map the system performance to gain sufficient situational awareness to facilitate mitigation efforts to eliminate or reduce undesired system behavior. This approach also enables differentiation of acceptance criteria to keep constrained specific performance and at the same time ensure a minimum performance throughout the parameter space which is possible and feasible. The proposed procedure also gives the company a solid rationale for what to test and not.

This research is limited in the way that we base our findings on case studies from one industrial case company in Norway. To address this limitation and improve the research further, we have identified the following possible future work; perform a case study utilizing the proposed procedure in practice and test the proposed procedure in multiple cases in various settings.

Acknowledgment

We would like to acknowledge the Norwegian Industrial Systems Engineering Research Group at USN for their contributions. Also, the case company for sharing relevant industry information through their SMEs.

References

[1] KONGSBERG. www.kongsberg.com (accessed 27th December, 2023).

[2] K. A. Kjeldaas, R. A. Haugen, and E. Syverud, "Challenges in Detecting Emergent Behavior in System Testing," in *INCOSE International Symposium 2021*, Virtual, 17-22 July 2021, vol. 31: Wiley, pp. 1211-1228, doi: 10.1002/j.2334-5837.2021.00896.x.

[3] O. Skreddernes, R. A. Haugen, and C. Haskins, "Coping with Verification in Complex Engineered Product Development," in *INCOSE International Symposium*, Hawaii, USA, 2023, vol. 33, no. 1: Wiley, pp. 482-502, doi: 10.1002/iis2.13034.

[4] R. A. Haugen, N.-O. Skeie, G. Muller, and E. Syverud, "Detecting emergence in engineered systems: a literature review and synthesis approach," *Systems Engineering*, vol. 26, no. 4, pp. 463-481, 2023, doi: 10.1002/sys.21660.

[5] R. A. Haugen and M. Mansouri, "Applying Systems Thinking to Frame and Explore a Test System for Product Verification; a Case Study in Large Defence Projects," in *INCOSE International Symposium 2020*, Virtual, 2020, vol. 30: Wiley, pp. 78-93, doi: 10.1002/j.2334-5837.2020.00709.x.

[6] R. A. Haugen and A. Ghaderi, "Modelling and Simulation of Detection Rates of Emergent Behaviors in System Integration Test Regimes," in *SIMS*, Virtual, 2021: Linköping Electronic Conference Proceedings, doi: 10.3384/ecp211858.

[7] R. A. Haugen, N.-O. Skeie, and G. Muller, "Automation Processes for Efficient Verification of Complex Systems: an Empirical Case Study," *Systems Engineering*, Submitted, doi: 10.13140/RG.2.2.32899.48166.

[8] R. A. Haugen, S. Kokkula, A. Ghaderi, G. Muller, and E. Syverud, "A Novel Approach to Detect Detrimental Weak Emergent Behavior Considering Noise Parameters: an Autonomous Search and Rescue Case Study," *IEEE Systems*, Submitted, doi: 10.36227/techrxiv.170491502.20215367/v1.

[9] G. Muller, "Industry-as-Laboratory Applied in Practice: The Boderc Project," in *gaudisite*, ed, 2004.

[10] W. P. M. Heemels and G. Muller, "Five Years of Multi-Disciplinary Academic and Industrial Research; Lessons Learned," in *gaudisite*, ed, 2006.

[11] Incose and Wiley, Fifth, Ed. *INCOSE Systems Engineering Handbook*. New York: John Wiley & Sons, Incorporated, 2023.

[12] SEBoK, *Measurements*, E. i. C. Robert Cloutier, ed.: San Diego, CA: International Council on Systems Engineering (INCOSE), 2023. [Online]. Available: https://sebokwiki.org/wiki/Measurement. Accessed on: 20th November 2023.

[13] K. Dunn, *Process Improvements Using Data*. 2021.

[14] D. C. Montgomery, *Design and Analysis of Experiments, 8th ed.* Wiley, 2017.

[15] S. Mittal, S. Diallo, and A. Tolk, *Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach*. Wiley (in English), 2018, pp. 1-395.

[16] H. Kopetz, A. Bondavalli, F. Brancati, B. Frömel, O. Höftberger, and S. Iacob, "Emergence in Cyber-Physical Systems-of-Systems (CPSoSs)," in *Cyber-Physical Systems of Systems: Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy*, A. Bondavalli, S. Bouchenak, and H. Kopetz Eds. Cham: Springer International Publishing, 2016, pp. 73-96.

[17] S. Y. Diallo, C. J. Lynch, R. Gore, and J. J. Padilla, "Emergent behavior identification within an agent-based model of the Ballistic Missile Defense System using statistical debugging," (in English), *J. Def. Model. Simul.-Appl. Methodol. Technol.-JDMS*, Article vol. 13, no. 3, pp. 275-289, 2015, doi: 10.1177/1548512915621973.

[18] A. K. Raz, J. Llinas, R. Mittu, and W. Lawless, "Engineering for Emergence in Information Fusion Systems: A Review of Some Challenges," in *22nd International Conference on Information Fusion, FUSION 2019*, Ottawa, Canada, 2-5 July, 2019 2019: Institute of Electrical and Electronics Engineers Inc.

[19] R. Raman and Y. Jeppu, "An Approach for Formal Verification of Machine Learning based Complex Systems," in *INCOSE International Symposium 2019*, Orlando, FL, USA, 20-25 July, 2019 2019, vol. 29, pp. 544-559, doi: 10.1002/j.2334-5837.2019.00620.x.

[20] J. Hsu, J. Clymer, J. Garcia Jr., and E. Gonzalez, "Agent-Based Modeling the Emergent Behavior of A System-of-Systems," *INCOSE International Symposium*, vol. 19, no. 1, pp. 1581-1590, 2009, doi: 10.1002/j.2334-5837.2009.tb01036.x.

[21] W. K. V. Chan, "Interaction metric of emergent behaviors in agent-based simulation," in *Proceedings of the 2011 Winter Simulation Conference (WSC)*, Phoenix Arizona, December 11 - 14, 2011 2011, pp. 357-368, doi: 10.1109/WSC.2011.6147763.

[22] C. Guariniello, R. A. Khalid, Z. Fang, and D. Delaurentis, "System-of-systems tools and techniques for the analysis of cyber-physical systems," *Systems Engineering*, vol. 23, no. 4, pp. 480-491, 2020, doi: 10.1002/sys.21539.

[23] M. A. Cummings, "Identifying and quantifying emergent behavior through system of systems modeling and simulation," 2015.

[24] Minitab. "Minitab." www.minitab.com (accessed 29th March, 2023).
[25] R. K. Roy, E. J. Kehoe, Ed. *A Primer on the Taguchi Method*, 2nd ed. Michigan, USA: Society of Manufacturing Engineers (in English), 2010, p. 304.
[26] K. Giammarco, "Practical modeling concepts for engineering emergence in systems of systems," in *12th System of Systems Engineering Conference (SoSE)*, Waikoloa, HI, USA, 18-21 June 2017, pp. 1-6, doi: 10.1109/SYSOSE.2017.7994977.
[27] S. D. Harris and J. M. Narkevicius, "Emergent Failure Modes and What to Do about Them," *INCOSE International Symposium*, vol. 26, no. 1, pp. 1044-1058, 2016, doi: 10.1002/j.2334-5837.2016.00210.x.