ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

Zero-Trust for the System Design Lifecycle

Douglas L. Van Bossuyt[1]
Department of Systems Engineering,
Naval Postgraduate School,
Monterey, CA 93943
e-mail: douglas.vanbossuyt@nps.edu

Britta Hale
Department of Computer Science,
Naval Postgraduate School,
Monterey, CA 93943
e-mail: britta.hale@nps.edu

Ryan Arlitt
Department of Mechanical Engineering,
Technical University of Denmark,
2800 Kongens Lyngby, Denmark
e-mail: rmarl@dtu.dk

Nikolaos Papakonstantinou
VTT Technical Research Centre,
FI-02044 Espoo, Finland
e-mail: Nikolaos.Papakonstantinou@vtt.fi

In an age of worsening global threat landscape and accelerating uncertainty, the design and manufacture of systems must increase resilience and robustness across both the system itself and the entire systems design process. We generally trust our colleagues after initial clearance/background checks; and systems to function as intended and within operating parameters after safety engineering review, verification, validation, and/or system qualification testing. This approach has led to increased insider threat impacts; thus, we suggest moving to the "trust, but verify" approach embodied by the Zero-Trust paradigm. Zero-Trust is increasingly adopted for network security but has not seen wide adoption in systems design and operation. Achieving the goal of Zero-Trust throughout the systems lifecycle will help to ensure that no single bad actor—whether human or machine learning/artificial intelligence (ML/AI)—can induce failure anywhere in a system's lifecycle. Additionally, while ML/AI and their associated risks are already entrenched within the operations phase of many systems' lifecycles, ML/AI is gaining traction during the design phase. For example, generative design algorithms are increasingly popular, but there is less understanding of potential risks. Adopting the Zero-Trust philosophy helps ensure robust and resilient design, manufacture, operations, maintenance, upgrade, and disposal of systems. We outline the rewards and challenges of implementing Zero-Trust and propose the framework for Zero-Trust for the system design lifecycle. This article highlights several areas of ongoing research with focus on high priority areas where the community should focus efforts. [DOI: 10.1115/1.4062597]

Keywords: artificial intelligence, cyber physical security for factories, cyber physical system design and operation, information management, machine learning for engineering applications, model-based systems engineering, Zero-Trust, systems engineering, system design

1 Introduction

The last several decades have seen significant advances in the design, manufacture, and operation of systems (the system design lifecycle). In today's global integrated environment, design teams can be distributed across multiple time zones and countries, manufacturing can be done a half a world away from raw materials and from consumers, and operating large, complex systems requires many people in many locations. While 100 years ago, a locomotive might be entirely manufactured in one industrial complex, and with raw materials sourced regionally, now such endeavors are intercontinental and involve a complex logistics and supply chain. Over the intervening decades, the threats that can be posed to the system design lifecycle have also become more complex. For instance, while industrial espionage generally required physical access in prior decades, today, a company's entire collection of product information can be stolen via remotely executed cyber attacks. However, over the same time period, the response to such threats has not kept pace. Oftentimes, after an initial background investigation, an employee is trusted without further oversight. Supply chains can similarly have weaknesses where counterfeit or defective parts can be introduced after initial vendor certification, or during repair and update processes. The introduction of machine learning/artificial intelligence (ML/AI) into the system design lifecycle further complicates matters as there are many new ML/AI attack vectors that organizations have likely not previously experienced.

This position article introduces the concept of Zero-Trust for the system design lifecycle to provide a framework (the framework for Zero-Trust for the system design lifecycle) and call to action for the research community to mitigate the complex and evolving threat landscape so that systems are designed, manufactured, and operated safely and securely. We believe that the framework will help to guide researchers in developing new methods and updating existing methods that are necessary to realize the vision. The rest of this article includes background information on the system design lifecycle and Zero-Trust, an overview of the proposed framework, a review of work conducted to date that we are aware of, a simple example of applying Zero-Trust to a system design lifecycle, and a call to action for future research directions in the community, and concludes with remarks on the future threat landscape.

2 Background

2.1 The System Design Lifecycle. The system design lifecycle is the process by which a system is conceptualized, designed,

[1]Corresponding author.

Manuscript received January 17, 2023; final manuscript received May 16, 2023; published online June 9, 2023. Assoc. Editor: Satyandra K. Gupta.

manufactured, tested, operated and used, maintained, upgraded, and eventually disposed of at the end of its useful life. Many different fields of engineering have preferences for specific lifecycle models and processes such as in mechanical engineering with the product design lifecycle [1,2], software engineering [3,4] to include DevOps [5], and life cycle engineering in industry 4.0 [6].

We use the systems engineering version of the system design lifecycle because it can accommodate most other engineering disciplines by design. Many different models such as the V-model, the spiral model, the waterfall model, various agile methods,[2] and others describe the same general systems engineering system design lifecycle as implemented in a variety of different ways [8–10]. They all follow a generalized process best represented by the V-model, which starts with understanding customer needs and developing requirements; proceeds through conceptual design; moves into detailed design where subject-matter experts are engaged to conduct discipline-specific design work; then enters an implementation and/or manufacturing phase; a test, integration, and validation phase; a system validation and verification phase; an operations and maintenance phase; and eventually a disposal phase [8,11,12]. Figure 1 shows a typical V-model.

2.2 Zero-Trust. The concept of Zero-Trust comes from the Zero-Trust security model with the Russian proverb of "trust but verify" representing the underlying philosophy [13–15]. Zero-Trust assumes any component in or outside of a system could be faulty or compromised, including human operators. Fields including the Internet of things (IoT), Big Data, and Infrastructure as a Service have embraced Zero-Trust since its original conception in 1994 [13,16–19].

As a caution, Zero-Trust can be mistakenly interpreted as frequent authentication, or simply a duplicate, parallel security check to any possible point of attack. This can be seen in efforts that approach Zero-Trust through entity authentication, multi-factor authentication, or repeated authorization checks [20,21]. While there is value in select security controls in case of failure (e.g., multi-factor authentication), Zero-Trust does not simply refer to redundancy. Furthermore, authentication itself has been a pillar of security foundations, and simply ensuring proper authorization likewise does not imply that Zero-Trust has been achieved; otherwise, Zero-Trust would be a re-branding of known good practice. Even methods of continuous authorization [22] fail to capture the full intent of Zero-Trust, and can place heavy overhead on users. So what is the distinction, then, of Zero-Trust?

A true Zero-Trust approach takes into account possible failure or attack susceptibility of *every* component in the system, piece by piece; assesses the risk from possible vulnerabilities in each; implements mitigation measures where possible; and provides a documented implication traceability graph such that if a failure occurs in the future, all affected aspects can easily be identified. Thus, while access control methods feature strongly in Zero-Trust guidance documents, such documents also contain a multitude of other security measures and testing directives [23,24], which should not be overlooked.

This holistic approach to security takes on interesting nuances when the "system" is considered not only in terms of information protection or cyber security, but in terms of the aggregated dependencies present across engineering, human factors, cyber, etc. Yet even that understanding of the trustworthiness of a system is incomplete, as even that picture is incomplete and lacks a critical dimension—time. Systems necessarily change over time, including software updates, part replacements, rewiring and repairs, and reliable employees with access leaving, while new ones are hired to fill their roles. All of these things effect the *trustworthiness* of the system, and accounting for them is inherent in an effective Zero-Trust approach to the system design lifecycle.

2.3 Emerging Technologies Impacting the System Design Lifecycle. Many emerging technologies and methodologies are rapidly gaining traction across the entire system design lifecycle that hold the promise of better, faster, and cheaper systems. For instance, foundation models such as transformers, diffusion models, and generative adversarial networks (GANs) have recently gained significant research attention [25,26] and enormous popular interest [27–31] for applications ranging from creative design to code generation [32–34].

For creative endeavors, this has prompted debate on the meaning, value, and ownership of art [35,36]. In contexts where correctness is important, the propensity of population foundation models to confidently hallucinate misinformation [26,37,38] and ongoing efforts to impose correctness on generated results [39,40] demonstrate a key vulnerability in practice. Not only must the training data be trusted but also the model and the correctness authority must be trusted.

An emerging technology that impacts operation of systems is the infusion of ML/AI into control of systems including when systems with major ML/AI components work as part of a human–machine team [41–45]. A variety of well-known and emerging threats to ML/AI exist; however, we have observed disconnects between systems designers, software developers, systems integrators, operators, and maintenance crews when using systems with significant ML/AI components such as uncrewed vehicles (UxSs) including uncrewed aerial vehicles, uncrewed ground vehicles, uncrewed surface vessels, and uncrewed underwater vehicles [41]. Less complex systems with ML/AI such as nuclear reactor spent fuel cooling pools can also be vulnerable [46,47]. While aspects of zero-trust have started to be adopted in portions the system design lifecycle for systems containing ML/AI components, we have not observed any frameworks to ensure Zero-Trust is imbued throughout the lifecycle and in all aspects of the lifecycle.

3 A Proposed Framework for Zero-Trust for the System Design Lifecycle

We propose that the community adopt the framework for Zero-Trust for the system design lifecycle. The goal of this framework is to ensure that all aspects of the system design lifecycle adhere to the Zero-Trust philosophy and that different phases and aspects of the system design lifecycle have their individual Zero-Trust

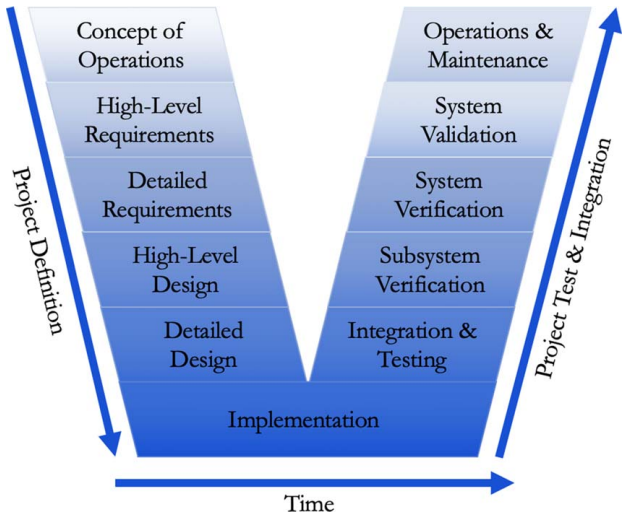

Fig. 1 A typical V-model of the systems engineering system design lifecycle

[2]Some argue that agile and related methods are fundamentally different than other system design lifecycle methods. We posit that at their core, agile methods have similar phases and processes as other system design lifecycle methods and have the same broad goals [7].

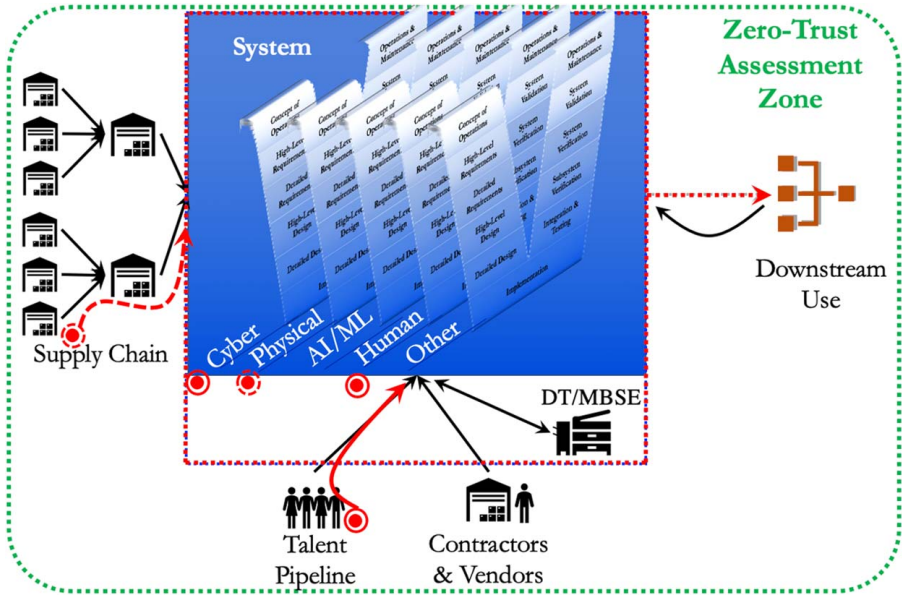

Fig. 2 A model of the proposed framework for Zero-Trust for the system design lifecycle in the example scenario discussed in Sec. 6. The points surrounded by solid and dashed circles indicate points of adversarial inject, while dotted points and lines indicate injects successfully mitigated by local Zero-Trust measures. Given the lack of continuous mitigation on the talent pipeline and only partial measures implemented for cybersecurity, both cyber and human system lifecycle V-models are flawed. The end result is an adversarially poisoned system.

elements linked together. In a sense, we are proposing a transition for the system design lifecycle that is similar to the transition that is occurring in the systems engineering community following the introduction of model-based systems engineering (MBSE) and currently with the ongoing adoption of MBSE [48–50]. In fact, the adoption of MBSE and digital twin (DT) (part of digital thread and digital engineering) [51,52] introduces new avenues of vulnerability in the system design lifecycle that we posit security measures alone cannot address.

Thus, we suggest that all aspects of the system design lifecycle must have Zero-Trust as an integral part of all processes, decisions, and so on. For instance, proper vetting and monitoring of all personnel involved in developing concepts of operations for new systems must be conducted to ensure the concepts of operations are not poisoned. Similarly, high-level and detailed requirements must be developed by trusted personnel under a Zero-Trust philosophy. Indeed, on critical systems with disastrous outcomes for system failures such as nuclear reactors or fighter jets, adopting parallel firewalled design teams (both conceptual and detailed) is appropriate to check the designs of the multiple teams against each other. Through our professional practice, we have already observed this behavior in several industries such as with design activities for safety-critical control systems where multiple firewalled design teams develop control systems that are then integrated into a larger voting block logic control system.

Other aspects of the system design lifecycle must also be infused with Zero-Trust such as the supply chain from raw material to components to subassemblies and finally to a completed system. Many opportunities exist throughout the supply chain for counterfeit materials and parts, for instance, to be introduced with potentially disastrous results [53–55]. While counterfeit parts have been an issue for many years, research is ongoing in detecting, mitigating, and accounting for counterfeiting in the supply chain.

We conceptualize the framework for Zero-Trust for the system design lifecycle as including the physical, cyber, AI/ML, and human domains of the system lifecycle. Systems and enterprise architectural models [9,56] can be useful in identifying potentially overlooked elements of these three domains. Functional hierarchy models, function flow block diagrams, physical hierarchies, component diagrams, and similar can be used to identify how people, information, material, and data flow through a system and a system design process. A variety of ontologies are available to help with such endeavors such as the functional basis for engineering design [57,58] and the energy, matter, material wealth, and information ontology [59,60].

Figure 2 graphically shows the proposed framework for Zero-Trust for the system design lifecycle, superimposed with a threat case (dashed line on the left, curved line on the bottom, circled dots, dashed box in the center of the image) that will be discussed in the example in Sec. 6. Within the Zero-Trust assessment, the scope falls not only on aspects of the system itself, spanning from physical to cyber and human aspects, but also on the inputs to that system. Since adversarial effects can propagate not only on physical dependencies (e.g., computer chips used in other subsidiary components) but also within standard software and AI/ML, inputs thus include the supply beyond the immediate suppliers, human talent pipeline, and contractors or vendors who may be involved in system design or access, repair, or who maintain system components. Furthermore, downstream systems can feed inputs back to the system in question, such as in federated learning, requirements definitions, and parts return. When DT/MBSE is used, it can further affect the system, even as the system itself gives inputs to the DT/MBSE. Thus, adversarial injects in the supply chain can propagate to the system and thence to the DT/MBSE, which can then adversely reflect back to poison the system itself even if the system vulnerability has been repaired in the interim. Thus, the Zero-Trust assessment zone wraps around not only the core system but also the greater system of dependencies.

We note that while critical systems such as nuclear reactors and aircraft carriers should use Zero-Trust throughout the system design lifecycle, other systems such as non-IoT toasters and decorative vases do not need as significant of an investment in Zero-Trust. We suggest that an assessment should be conducted at the start of a new system design lifecycle to determine the "right" amount of Zero-Trust to infuse throughout the lifecycle. This may include limiting where Zero-Trust is applied or how thoroughly Zero-Trust is implemented.

4 Zero-Trust Application Work to Date

Various works have looked at Zero-Trust for systems engineering [46,61], with some extending to consideration of AI/ML components [41,47]. Still other work has applied Zero-Trust to creating a supply chain framework [62]. Under software engineering, investigation of Zero-Trust application to both general software, DevSecOps, and Systems Theoretic Process Approach principals has been made [63–66], yet consideration of the full system lifecycle—whether a typical hardware and software integrated system (a cyber physical system), or simply software system lifecycle—has remained largely unconsidered.

Interestingly, research into AI/ML specifically has teetered on the edge of Zero-Trust for the design lifecycle. In AI/ML, model retraining must sometimes occur, which raises the question of possible effects of data poisoning for the data used either under the initial training or later retraining phases. Methods of exploitation of this retraining environment have been investigated [67,68], indicating that the reliability and trustworthiness of AI/ML over time should be a concern for system designers. However, while general poisoning detection has received attention [69,70], an overarching design lifecycle and testing concept is new to this article.

The supply chain has begun to receive Zero-Trust attention and a mapping has been produced for the supply chain to show what Zero-Trust concepts can be used where [62]. The cyber supply chain has also started to be infused with Zero-Trust concepts although much work remains to be done [71]. Portions of the critical infrastructure sectors have begun to apply cyber supply chain Zero-Trust techniques [72]. However, wide-scale adoption throughout the supply chain remains elusive and will require significant additional research. Manufacturing, an activity that can be considered a sub-set of the supply chain, is also starting to see interest in Zero-Trust [73,74].

Prognostics and health management (PHM) of systems has begun to see Zero-Trust concepts proposed for integration into existing methods and processes [75,76]. Some applications have been explored such as Zero-Trust for PHM of distributed engine controls [77]. There remains many fertile areas within PHM for further research into integrating Zero-Trust.

Several proposals and demonstration implementations have been done for Zero-Trust in DTs. An exploration for smart grid applications was conducted [78]. Smart city DT researchers have considered Zero-Trust [79]. Other research has suggested Zero-Trust for DTs in a variety of applications such as in certain systems engineering contexts [80].

As evidenced earlier, many researchers and practitioners are beginning to adopt the Zero-Trust philosophy for a variety of specific applications. However, these efforts have not been coordinated in a way that leads to easy integration into the framework for Zero-Trust for the system design lifecycle. Much work remains to be done to integrate these disparate efforts and develop new Zero-Trust approaches for portions of the system design lifecycle that have yet to receive the Zero-Trust treatment.

5 Proposed Future Directions and a Call to Action

We propose to the community that Zero-Trust be a consideration in the development of new methods, ontologies, schemata, processes, and techniques for all aspects of the system design lifecycle. We also suggest that Zero-Trust be applied to all aspects of our research endeavors including our computer systems; our note-taking devices (physical and cyber) such as notebooks; our personnel including our collaborators and our graduate students; the peer review process; and our manuscript development, publication, and dissemination processes. While this may seem extreme, graduate students have been identified as intelligence assets for a variety of governments [81], for example.

The human element in our research endeavors deserves further thought and consideration. As a community, we may wish to develop a security posture similar to that of many national defense agencies where we are more cautious with our conversations and collaborations. For instance, conferences can be exploited by malicious actors in a variety of ways such as suggesting specific approaches with inherent flaws; becoming a trusted collaborator to an established and trustworthy research group, and introducing intentionally bad data into ML/AI training sets; and other similar activities. However, we will need to strike a balance as a community between caution and openness lest we fall victim to xenophobia, for instance.

An interface standard and ontology must be developed to aid in passing Zero-Trust information between different portions of the system design lifecycle and between the different layers. This will help to ensure that Zero-Trust is applied throughout the lifecycle, and gaps in coverage are minimized. However, thought must also be given to how Zero-Trust data are protected to prevent malicious actors from using a Zero-Trust database to identify exploitable weaknesses. This may result in a new class of interface control diagram [82] specifically to negotiate Zero-Trust issues between the contractors developing subsystems, for instance.

The importance of knowledge reuse to the systems design lifecycle is well documented [57,83]. Many ontologies have been developed to aid in knowledge capture and knowledge reuse [84]. A framework and ontology will be needed that directly addresses Zero-Trust. Similarly, a Zero-Trust paradigm for a system design lifecycle will need to account for the provenance of information coming from a knowledge reuse source such as a product database or a design repository [85].

Another issue that must be addressed is how we will verify and validate (V&V) that new research developments adhere to the Zero-Trust philosophy. New methods of conducting V&V specifically for Zero-Trust must be developed. Further, methods of ensuring that these Zero-Trust V&V methods themselves are not compromised must be identified and implemented. The same data poisoning problems that can plague ML/AI development could exist in V&V of Zero-Trust across our research.

As a starting point, we urge the community to specifically investigate Zero-Trust for DTs and MBSE. As DTs and MBSE become a more integral part of new and existing systems, their cross-cutting nature throughout the system design lifecycle [51] makes them a particularly tempting target. Getting Zero-Trust "right" here will have a virtuous self-reinforcing effect across other layers of the system design lifecycle.

Another area we believe needs immediate attention is the command and control (C2), and the broader command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) of ML/AI component-containing systems. While great effort has gone into securing C2 datalinks and C4ISR modules for UxS in defense applications, for instance, significant work remains to adhere to the Zero-Trust philosophy. As more UxS are fielded in more industries such as last mile delivery [86], global shipping [87], and agriculture [88], the imperative grows to ensure that Zero-Trust is infused into C2 and C4ISR.

The twin issues of how to efficiently address product complexity team size in a Zero-Trust paradigm must be examined, while all other Zero-Trust developmental work is carried out. While Zero-Trust for various stages of the system design lifecycle can be implemented on small systems and with small teams, the overhead necessary to infuse Zero-Trust across an entire system design lifecycle for a very complex system with thousands of engineers could prove too costly in terms of time, resources, and computational power to implement. Thus, it is vitally important that the community ensure low overhead is needed even on extremely complex systems such as nuclear reactors, aircraft carriers, and similar.

The rapid rise of GANs, large language models (LLMs), computational design tools based on AI/ML, and other AI systems indicates a need for the community to consider these while developing Zero-Trust system design lifecycle methods. Computational challenges already exist with LLMs and GANs, and are likely

to only increase when a Zero-Trust paradigm is implemented. In some instances, it may be too computationally intensive to implement a Zero-Trust paradigm, and instead adaptations will be needed in system design lifecycle methods to assume that GANs and LLMs are always compromised.

Many other areas should be identified by the research community as needing immediate attention. Every aspect of the system design lifecycle must be examined to determine what needs to be addressed within a Zero-Trust framework. A plan must be developed on how to prioritize research into Zero-Trust for the system design lifecycle that includes directing funding from major agencies toward areas of greatest need. Perhaps, the National Science Foundation or the Defense Advanced Research Projects Agency may sponsor a planning workshop for Zero-Trust across the system design lifecycle.

6 An Example: Zero-Trust for the System Design Lifecycle of a 3D Printer

This section presents a brief vignette of how the framework for Zero-Trust for the system design lifecycle could be implemented. A simple fused filament fabrication 3D printer [89,90] is used as the example system. The 3D printer is being constructed for use in austere environments and builds upon existing 3D printer heritage [91]. Figure 2 graphically illustrates the example.

A company with significant heritage in 3D printer design, manufacture, and operations has identified a potential market opportunity based on a concept of operations released to the public from a national defence organization. The company does due diligence to investigate the origin of the concept of operations to ensure that it is credible, accurate, and representative of the national defense organization's needs. This adheres to the "trust, but verify" mantra of Zero-Trust.

The company assigns a team consisting of a systems engineer, several subject-matter experts, and associated engineering support staff to develop high-level requirements. While the company originally conducted criminal background checks as part of the hiring process on its employees, many of the personnel assigned to the project have been with the company for a number of years and have implicit trust throughout the organization. The company does not have a monitoring program for its employees to verify that they remain trustworthy. Newer employees hired straight from college or graduate school have gone through criminal background checks and have generally been assumed to be trustworthy by their colleagues but remain unproven. Thus, the Zero-Trust philosophy has not been applied to the talent pipeline and the system design lifecycle is vulnerable in the human layer to malicious attacks.

Next, detailed requirements are developed and system architecture begins. The data systems of the company use two factor authentication, but employees have broad access to internal systems with little oversight. This presents a cyber security vulnerability if an outside actor manages to successfully steal credentials and access the systems. However, the threat is limited to a degree by the two factor authentication implementation. Insider threats remain a possibility. Thus, Zero-Trust is partially implemented in the cyber layer at this step in the system design lifecycle.

During system architectural efforts and initial high-level design, risk analysis methods that incorporate the Zero-Trust philosophy [41,46,47,80] are used extensively to identify potential exploit points in the detailed design and operation of the system. This successfully addresses most Zero-Trust concerns at this phase of the system design lifecycle although more methods are needed to fully identify potential risks that Zero-Trust could address.

As detailed design begins, subject-matter experts undertake the design of the hardware and software that will comprise the 3D printer. Because of the heritage that the organization has in developing 3D printers, a significant amount of reuse from prior designs can be undertaken. However, no method of verifying archived data has not been tampered with exists within the company. The subject-matter experts rely upon the existing archived data from a design repository, some of which they may have originally created, to conduct detailed design of the system. Further, they rely upon both open- and closed-source software for 3D printer control but do not vet the software code bases, instead relying upon their belief in the trustworthiness of the software supply chains. The designers also employ experimental foundation models to conduct design iterations and exploration, and trust the results produced by the experimental foundation models without rigorously verifying that the models used did not contain exploits. This violates Zero-Trust principles and leaves the 3D printer vulnerable to a variety of attacks.

Now the first prototypes of the 3D printer are constructed. A production line including the logistics and supply chain is setup to begin production of components and subassemblies in advance of system V&V activities based upon the significant heritage of the company successfully introducing new 3D printer products. A method of ensuring Zero-Trust through the supply chain [62] is implemented within the company, which greatly reduces the potential of counterfeit parts and intentional defects being introduced. This is a successful application of Zero-Trust at this phase of the system design lifecycle and across the various layers.

During the V&V process, the company follows a standardized procedure to test various elements of the design at the subsystem and system level to ensure proper functionality. However, no effort is made to ensure data integrity or test validity because the company has been successful without needing to undertake these steps in the past. This opens a variety of vulnerabilities in the V&V process where a malicious actor could introduce faults into the system through faked test results and similar methods. Zero-Trust principles were not applied to this phase of the process.

Finally, the 3D printer is mass produced and sold to a national defense organization for use in austere environments. A maintenance and logistics supply chain is setup to enable the 3D printers to be repaired in the field and serviced at depots. ML/AI is used on PHM data collected from the 3D printers to determine when maintenance must occur. During this phase of the system design lifecycle, no special measures are taken to ensure Zero-Trust is implemented on the ML/AI component. However, the national defense organization does successfully implement Zero-Trust principles for the maintenance and logistics organization and associated personnel.

During a national security crisis, the 3D printers are deployed with national defense forces to secure a border region from an aggressive neighbor. Unfortunately, the myriad of vulnerabilities throughout the system design lifecycle presented opportunities for personnel from the aggressive neighboring country to leave several software and hardware exploits within the 3D printers that are activated during the national security crisis. This leads to some 3D printers being completely disabled, while others continue to print but introduce subtle defects into the parts produced, which cause those parts to fail at a much higher rate during use by the national security forces. Had a more thorough Zero-Trust approach been implemented throughout the system design lifecycle, this scenario may have been avoidable.

7 Conclusion

Ultimately, the security and reliability of system outputs rely on the security and reliability of the system itself. Many works have considered methods of expanding the security and reliability zone by not assuming trust in any given component, i.e., Zero-Trust. However, these have not accounted for system inputs and external dependencies, including later feedback loops. Expanding Zero-Trust to consider not only these but also the *time dimension* of systems supports a continuous monitoring and hardening against attacks. Securing a system at the point of design only leaves it open to the standard and realistic processes of adversarial injects that change and adapt over time.

Zero-Trust is a methodology process, not an end-state. Analysis of any given system may overlook vulnerabilities, patch vulnerabilities later in the future, and introduce faults, and a perfect solution is neither realistic nor the end goal. Instead, the intent of lifecycle Zero-Trust is to expand the zone of consideration, bringing into focus vulnerabilities hitherto overlooked in many system V&V efforts, facilitating improvement and system robustness to the ever-adapting adversarial methodology.

Thus, we have proposed the framework for Zero-Trust for the system design lifecycle as a means of focusing attention throughout the entire system design lifecycle on Zero-Trust principles to reduce vulnerability, increase security, and increase reliability of systems. We suggested analyzing several layers of the system design lifecycle including cyber, physical, AI/ML, human, and other layers. We identified several major potential threat vectors such as the supply chain, the talent pipeline, contractors and vendors, and the DT/MBSE environment as well as downstream use. While a significant body of research exists in the literature on Zero-Trust as applied to several portions of the system design lifecycle, many gaps still exist and a unified method of communicating Zero-Trust information between the various steps and layers of the system design lifecycle is not yet available. This position article is a call to action for the community to focus more effort on including Zero-Trust in all of our work moving forward so that we have safe, secure, and reliable systems.

Acknowledgment

Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators. Approved for Public Release; distribution is unlimited.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

No data, models, or code were generated or used for this paper.

References

[1] Otto, K. N., and Wood, K. L., 2001, *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River, NJ.

[2] Ullman, D., 2017, *The Mechanical Design Process*, David Ullman LLC, Redding, CA.

[3] Muller, M. J., Haslwanter, J. H., and Dayton, T., 1997, "Participatory Practices in the Software Lifecycle," *Handbook of Human-Computer Interaction*, M. G. Helander, T. K. Landauer, and P. V. Prabhu, eds., Elsevier, New York, NY, pp. 255–297.

[4] Ruparelia, N. B., 2010, "Software Development Lifecycle Models," ACM SIGSOFT Softw. Eng. Notes, **35**(3), pp. 8–13.

[5] Laukkarinen, T., Kuusinen, K., and Mikkonen, T., 2018, "Regulated Software Meets Devops," Inform. Softw. Technol., **97**, pp. 176–178.

[6] Stark, R., Grosser, H., Beckmann-Dobrev, B., and Kind, S., INPIKO Collaboration, 2014, "Advanced Technologies in Life Cycle Engineering," Procedia CIRP, **22**, pp. 3–14.

[7] Miller, A., Giachetti, R. E., and Van Bossuyt, D. L., 2022, "Challenges of Adopting Devops for the Combat Systems Development Environment," Defense AR J., **29**(1), pp. 22–49.

[8] Blanchard, B. S., Fabrycky, W. J., and Fabrycky, W. J., 2010, *Systems Engineering and Analysis*, Vol. 5, Prentice Hall, Englewood Cliffs, NJ.

[9] Crawley, E., Cameron, B., and Selva, D., 2015, *System Architecture: Strategy and Product Development for Complex Systems*, Prentice Hall Press, New York City, NY.

[10] Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D., and Shortell, T. M., eds., 2015, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th ed., Wiley, Hoboken, NJ.

[11] Forsberg, K., and Mooz, H., 1991, "The Relationship of System Engineering to the Project Cycle," INCOSE International Symposium, Chattanooga, TN, Oct. 21–23, pp. 57–65.

[12] Forsberg, K., and Mooz, H., 1992, "The Relationship of Systems Engineering to the Project Cycle," Eng. Manag. J., **4**(3), pp. 36–43.

[13] Marsh, S. P., 1994, "Formalising Trust as a Computational Concept," University of Stirling, Stirling, Scotland.

[14] Rose, S., Borchert, O., Mitchell, S., and Connelly, S., 2020, Zero Trust Architecture, Special Publication 800-207, National Institute of Standards and Technology, Gaithersburg, MD.

[15] Haber, M., 2020, *Privileged Attack Vectors: Building Effective Cyber-Defense Strategies to Protect Organizations*, 2nd ed., Springer, Berkeley, CA, pp. 295–304.

[16] Samaniego, M., and Deters, R., 2018, "Zero-Trust Hierarchical Management in IOT," IEEE International Congress on Internet of Things (ICIOT), San Francisco, CA, July 2–7, pp. 88–95.

[17] Tao, Y., Lei, Z., and Ruxiang, P., 2018, "Fine-Grained Big Data Security Method Based on Zero Trust Model," 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), Singapore, Dec. 11–13, pp. 1040–1045.

[18] Scott, B., 2018, "How a Zero Trust Approach Can Help to Secure Your AWS Environment," Netw. Section, **2018**(3), pp. 5–8.

[19] ACT-IAC Zero Trust Project Team, 2019, Zero Trust Cybersecurity Current Trends, American Council for Technology-Industry Advisory Council (ACT-IAC), https://www.actiac.org/zero-trust-cybersecurity-current-trends.

[20] Embrey, B., 2020, "The Top Three Factors Driving Zero Trust Adoption," Comput. Fraud Secur., **2020**(9), pp. 13–15.

[21] Scott, B., 2018, "How a Zero Trust Approach Can Help to Secure Your AWS Environment," Netw. Secur., **2018**(3), pp. 5–8.

[22] Dimitrakos, T., Dilshener, T., Kravtsov, A., La Marra, A., Martinelli, F., Rizos, A., Rosetti, A., and Saracino, A., 2020, "Trust Aware Continuous Authorization for Zero Trust in Consumer Internet of Things," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, Dec. 29–Jan. 1, IEEE, pp. 1801–1812.

[23] Young, S. D., 2021, "Moving the U.S. Government Toward Zero Trust Cybersecurity Principles," January, https://www.whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf, Accessed December 10, 2022.

[24] U.S. Department of Defense, 2021, "Department of Defense Releases Zero Trust Strategy and Roadmap," https://www.defense.gov/News/Releases/Release/Article/3225919/department-of-defense-releases-zero-trust-strategy-and-roadmap/, Accessed December 10, 2022.

[25] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A., 2018, "Generative Adversarial Networks: An Overview," IEEE Signal Process. Mag., **35**(1), pp. 53–65.

[26] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2020, "Generative Adversarial Networks," Commun. ACM, **63**(11), pp. 139–144.

[27] Marcus, G., Davis, E., and Aaronson, S., 2022, "A Very Preliminary Analysis of DALL-E 2," Preprint. arXiv preprint arXiv:2204.13807.

[28] Stöckl, A., 2022, "Evaluating a Synthetic Image Dataset Generated with Stable Diffusion," arXiv preprint arXiv:2211.01777.

[29] Borji, A., 2022, "Generated Faces in the Wild: Quantitative Comparison of Stable Diffusion, MidJourney and Dall-e 2," arXiv Preprint arXiv:2210.00586.

[30] Srivastava, M., 2023, "A Day in the Life of Chatgpt as a Researcher: Sustainable and Efficient Machine Learning—A Review of Sparsity Techniques and Future Research Directions," OSF Preprints.

[31] Sobania, D., Briesch, M., and Rothlauf, F., 2022, "Choose Your Programming Copilot: A Comparison of the Program Synthesis Performance of Github Copilot and Genetic Programming," Proceedings of the Genetic and Evolutionary Computation Conference, Boston, MA, July 9–13, pp. 1019–1027.

[32] Deverall, J., Lee, J., and Ayala, M., 2017, Using Generative Adversarial Networks to Design Shoes: The Preliminary Steps. Stanford University CS231n: Deep Learning for Computer Vision, Stanford, CA.

[33] Fan, J., Liu, T., Li, G., Chen, J., Shen, Y., and Du, X., 2020, "Relational Data Synthesis Using Generative Adversarial Networks: A Design Space Exploration," arXiv Preprint arXiv:2008.12763.

[34] Mao, Y., He, Q., and Zhao, X., 2020, "Designing Complex Architectured Materials With Generative Adversarial Networks," Sci. Adv., **6**(17), p. eaaz4169.

[35] Epstein, Z., Levine, S., Rand, D. G., and Rahwan, I., 2020, "Who Gets Credit for AI-Generated Art?," Iscience, **23**(9), p. 101515.

[36] Roose, K., 2022, "An AI-Generated Picture Won an Art Prize. Artists Aren't Happy," *The New York Times, 2*, p. 2022.

[37] Alqahtani, H., Kavakli-Thorne, M., and Kumar, G., 2021, "Applications of Generative Adversarial Networks (GANS): An Updated Review," Arch. Comput. Methods Eng., **28**(2), pp. 525–552.

[38] Huang, H., He, R., Sun, Z., and Tan, T., 2019, "Wavelet Domain Generative Adversarial Network for Multi-Scale Face Hallucination," Int. J. Comput. Vision, **127**(6), pp. 763–784.

[39] Shmelkov, K., Schmid, C., and Alahari, K., 2018, "How Good Is My Gan?" Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, Sept. 8–14, pp. 213–229.

[40] Cherian, A., and Sullivan, A., 2019, "Sem-Gan: Semantically-Consistent Image-to-Image Translation," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, Jan. 7–11, IEEE, pp. 1797–1806.

[41] Hale, B., Van Bossuyt, D. L., Papakonstantinou, N., and O'Halloran, B., 2021, "A Zero-Trust Methodology for Security of Complex Systems With Machine Learning Components," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference,

Online, Virtual, Aug. 17–19, Vol. 85376, American Society of Mechanical Engineers, p. V002T02A067.
[42] Matsuyama, L., Zimmerman, R., Eaton, C., Weger, K., Mesmer, B., Tenhundfeld, N., Van Bossuyt, D., and Semmens, R., 2021, "Determinants That Are Believed to Influence the Acceptance and Adoption of Mission Critical Autonomous Systems," AIAA Scitech 2021 Forum, Virtual, Jan. 11–15 and 19–21, p. 1156.
[43] Flynn, M., Smitherman, H. M., Weger, K., Mesmer, B., Semmens, R., Van Bossuyt, D., and Tenhundfeld, N. L., 2021, "Incentive Mechanisms for Acceptance and Adoption of Automated Systems," 2021 Systems and Information Engineering Design Symposium (SIEDS), Virtual, Apr. 30, IEEE, pp. 1–6.
[44] Weger, K., Matsuyama, L., Zimmermann, R., Mesmer, B., Van Bossuyt, D., Semmens, R., and Eaton, C., 2023, "Insight Into User Acceptance and Adoption of Autonomous Systems in Mission Critical Environments," Int. J. Hum. Comput. Interact., **39**(7), pp. 1423–1437.
[45] Schwalb, J., Menon, V., Tenhundfeld, N., Weger, K., Mesmer, B., and Gholston, S., 2022, "A Study of Drone-Based AI for Enhanced Human-AI Trust and Informed Decision Making in Human-AI Interactive Virtual Environments," IEEE 3rd International Conference on Human-Machine Systems (ICHMS), Orlando, FL, Nov. 17–19, pp. 1–6.
[46] Papakonstantinou, N., Van Bossuyt, D. L., Linnosmaa, J., Hale, B., and O'Halloran, B., 2021, "A Zero Trust Hybrid Security and Safety Risk Analysis Method," ASME J. Comput. Inf. Sci. Eng., **21**(5), p. 050907.
[47] Papakonstantinou, N., Hale, B., Linnosmaa, J., Salonen, J., and Van Bossuyt, D. L., 2022, "Model Driven Engineering for Resilience of Systems With Black Box and AI-Based Components," Annual Reliability and Maintainability Symposium (RAMS), Tucson, AZ, Jan. 24–27.
[48] Estefan, J. A., 2007, "Survey of Model-Based Systems Engineering (mbse) Methodologies," Incose MBSE Focus Group, **25**(8), pp. 1–12.
[49] Long, D., and Scott, Z., 2012, *A Primer for Model-Based Systems Engineering*, Lulu. com.
[50] Friedenthal, S., Moore, A., and Steiner, R., 2014, *A Practical Guide to SysML: The Systems Modeling Language*, Morgan Kaufmann, Burlington, MA.
[51] Bickford, J., Van Bossuyt, D. L., Beery, P., and Pollman, A., 2020, "Operationalizing Digital Twins Through Model-Based Systems Engineering Methods," Syst. Eng., **23**(6), pp. 724–750.
[52] Lee, E. B. K., Van Bossuyt, D. L., and Bickford, J. F., 2021, "Digital Twin-Enabled Decision Support in Mission Engineering and Route Planning," Systems, **9**(4), p. 82.
[53] Guin, U., Huang, K., DiMase, D., Carulli, J. M., Tehranipoor, M., and Makris, Y., 2014, "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," Proc. IEEE, **102**(8), pp. 1207–1228.
[54] Zhang, J., and Ge, M., 2011, "A Study of an Anti-Counterfeiting Fiber With Spectral Fingerprint Characteristics," J. Textile I nst., **102**(9), pp. 767–773.
[55] Stradley, J., and Karraker, D., 2006, "The Electronic Part Supply Chain and Risks of Counterfeit Parts in Defense Applications," IEEE Trans. Compon. Packaging Manuf. Technol., **29**(3), pp. 703–705.
[56] Giachetti, R., 2016, *Design of Enterprise Systems: Theory, Architecture, and Methods*, CRC Press, Boca Raton, FL.
[57] Stone, R. B., and Wood, K. L., 1999, "Development of a Functional Basis for Design," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, NV, Sept. 12–16, Vol. 19739, American Society of Mechanical Engineers, pp. 261–275.
[58] Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L., 2002, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," Res. Eng. Des., **13**(2), pp. 65–82.
[59] Langford, G. O., 2012, *Engineering Systems Integration: Theory, Metrics, and Methods*, CRC Press, Boca Raton, FL.
[60] Langford, G., and Langford, T., 2017, "The Making of a System of Systems: Ontology Reveals the True Nature of Emergence," 2017 12th System of Systems Engineering Conference (SoSE), Waikoloa, HI, June 18–21, IEEE, pp. 1–5.
[61] Papakonstantinou, N., Van Bossuyt, D. L., Linnosmaa, J., Hale, B., and O'Halloran, B., 2020, "Towards a Zero Trust Hybrid Security and Safety Risk Analysis Method," ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual/Online, Aug. 17–19.
[62] Collier, Z. A., and Sarkis, J., 2021, "The Zero Trust Supply Chain: Managing Supply Chain Risk in the Absence of Trust," Int. J. Prod. Res., **59**(11), pp. 3430–3445.
[63] Sanders, G., Morrow, T., Richmond, N., and Woody, C., 2021, Integrating Zero Trust and DevSecOps. Accession Number: AD1145432. Carnegie Mellon University. Pittsburgh, PA.
[64] Giray, G., 2021, "A Software Engineering Perspective on Engineering Machine Learning Systems: State of the Art and Challenges," J. Syst. Softw., **180**, p. 111031.
[65] Ishimatsu, T., Leveson, N. G., Thomas, J., Katahira, M., Miyamoto, Y., and Nakao, H., 2010, "Modeling and Hazard Analysis Using STPA," Proceedings of the 4th International Association for the Advancement of Space Safety (IAASS), Huntsville, AL, May 19–21.
[66] Abdulkhaleq, A., Wagner, S., and Leveson, N., 2015, "A Comprehensive Safety Engineering Approach for Software-Intensive Systems Based on STPA," *Procedia. Eng.*, Vol. 128, Elsevier, Amsterdam, Netherlands, pp. 2–11.
[67] Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W., and Zhang, X., 2018, "Trojaning Attack on Neural Networks," Network and Distributed System Security Symposium, San Diego, CA, Feb. 18–21.
[68] Shejwalkar, V., and Houmansadr, A., 2021, "Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning," Network and Distributed System Security Symposium, Virtual, Feb. 21–25.
[69] Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C., and Nepal, S., 2019, "Strip: a Defence Against Trojan Attacks on Deep Neural Networks," Proceedings of the 35th Annual Computer Security Applications Conference, San Juan, Puerto Rico, Dec. 9–13.
[70] Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., and Zhao, B. Y., 2019, "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, May 20–22, pp. 707–723.
[71] do Amaral, T. M. S., and Gondim, J. J. C., 2021, "Integrating Zero Trust in the Cyber Supply Chain Security," 2021 Workshop on Communication Networks and Power Systems (WCNPS), Fortaleza, Brazil, Nov. 17–18, IEEE, pp. 1–6.
[72] Aarland, M., and Gjøsæter, T., 2022, "Digital Supply Chain Vulnerabilities in Critical Infrastructure: A Systematic Literature Review on Cybersecurity in the Energy Sector," International Conference on Information Systems Security and Privacy (ICISSP), Virtual/Online, Feb. 9–11, pp. 326–333.
[73] Paul, B., and Rao, M., 2022, "Zero-Trust Model for Smart Manufacturing Industry," Appl. Sci., **13**(1), p. 221.
[74] Buras, B., Xanthopoulos, C., Butler, K., and Kim, J., 2022, "Zero Trust Approach to IC Manufacturing and Testing," 2022 IEEE International Test Conference (ITC), Anaheim, CA, Sept. 23–30, IEEE, pp. 583–586.
[75] Khemani, V., Azarian, M. H., and Pecht, M. G., 2021, "Prognostics and Secure Health Management of Electronic Systems in a Zero-Trust Environment," Annual Conference of the PHM Society, Vol. 13.
[76] Mao, Y., Ma, Z., Gao, S., Li, L., Yuan, B., Chai, B., He, P., and Liu, X., 2022, "A Method of Embedded Computer Degradation Trend Prediction," 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Chengdu, China, Aug. 19–21, IEEE, pp. 1338–1343.
[77] Pakmehr, M., Khamvilai, T., Behbahani, A. R., Costello, J., Skertic, R., and Ademola, A. P., 2022, "Applying Zero Trust Principles to Distributed Embedded Engine Control Systems," AIAA AVIATION 2022 Forum, Chicago, IL, June 27–July 1, p. 3480.
[78] Sellitto, G. P., Aranha, H., Masi, M., and Pavleska, T., 2021, "Enabling a Zero Trust Architecture in Smart Grids Through a Digital Twin," European Dependable Computing Conference, Munich, Germany, Sept. 13–16, Springer, pp. 73–81.
[79] Kismul, A., Al-Khateeb, H., and Jahankhani, H., 2023, "A Critical Review of Digital Twin Confidentiality in a Smart City," *Cybersecurity in the Age of Smart Societies*, H. Jahankhani, ed., Proceedings of the 14th International Conference on Global Security, Safety and Sustainability, London, pp. 437–450.
[80] Van Bossuyt, D. L., Hale, B., Arlitt, R. M., and Papakonstantinou, N., 2022, "Multi-mission Engineering With Zero Trust: A Modeling Methodology and Application to Contested Offshore Wind Farms," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, St. Louis, MO, Aug. 14–17, Vol. 86212, American Society of Mechanical Engineers, p. V002T02A058.
[81] Golden, D., 2017, *Spy Schools: How the CIA, FBI, and Foreign Intelligence Secretly Exploit America's Universities*, Henry Holt and Company, New York.
[82] Rigby, K. A., 2013, *Aircraft Systems Integration of Air-Launched Weapons*, John Wiley & Sons, Hoboken, NJ.
[83] Baxter, D., Gao, J., Case, K., Harding, J., Young, B., Cochrane, S., and Dani, S., 2007, "An Engineering Design Knowledge Reuse Methodology Using Process Modelling," Res. Eng. Des., **18**(1), pp. 37–48.
[84] Yang, L., Cormican, K., and Yu, M., 2019, "Ontology-Based Systems Engineering: A State-of-the-Art Review," Comput. Indust., **111**, pp. 148–171.
[85] Bohm, M. R., Stone, R. B., and Szykman, S., 2005, "Enhancing Virtual Product Representations for Advanced Design Repository Systems," ASME JCISE, **5**(4), pp. 360–372.
[86] Hoffmann, T., and Prause, G., 2018, "On the Regulatory Framework for Last-Mile Delivery Robots," Machines, **6**(3), p. 33.
[87] Levander, O., 2017, "Autonomous Ships on the High Seas," IEEE Spectrum, **54**(2), pp. 26–31.
[88] Mogili, U. R., and Deepak, B., 2018, "Review on Application of Drone Systems in Precision Agriculture," Procedia Comput. Sci., **133**, pp. 502–509.
[89] Singh, S., Singh, G., Prakash, C., and Ramakrishna, S., 2020, "Current Status and Future Directions of Fused Filament Fabrication," J. Manuf. Process., **55**, pp. 288–306.
[90] Steuben, J., Van Bossuyt, D. L., and Turner, C., 2015, "Design for Fused Filament Fabrication Additive Manufacturing," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Boston, MA, Aug. 2–5, Vol. 57113, American Society of Mechanical Engineers, p. V004T05A050.
[91] Jones, R., Haufe, P., Sells, E., Iravani, P., Olliver, V., Palmer, C., and Bowyer, A., 2011, "Reprap–the Replicating Rapid Prototyper," Robotica, **29**(1), pp. 177–191.