31st Annual INCOSE international symposium
Honolulu, HI, USA
July 17 - 22, 2021

Security in the Future of Systems Engineering (FuSE), a Roadmap of Foundation Concepts

Rick Dove
Paradigm Shift International
dove@parshift.com

Keith Willett
U.S. Department of Defense
kwillett@ctntechnologies.com

Tom McDermott
Systems Engineering Research Center
tmcdermo@stevens.edu

Holly Dunlap
Raytheon Technologies
holly.dunlap@raytheon.com

Delia Pembrey MacNamara
Australian Government
deliamacnamara@gmail.com

Cory Ocker
Raytheon Technologies
Cory.L.Ocker@raytheon.com

Abstract. The Future of Systems Engineering (FuSE) is an INCOSE-led multiorganizational collaborative initiative pursuing INCOSE's *Vision 2025* and beyond. To accomplish this the FuSE initiative encompasses a number of topic areas with active projects to shape the future of systems engineering. This paper addresses the FuSE Security topic area and provides a roadmap of eleven foundational concepts for building the security vision. The purpose of this paper is to instigate and inspire thinking and involvement in the development and practice of the foundational concepts.

Introduction

The Future of Systems Engineering (FuSE) is an INCOSE led multi-organization collaborative initiative that has identified a number of specific project areas to be pursued (INCOSE n.d.). The principle purpose of FuSE is to realize the INCOSE vision for the future of systems engineering. The authors of this paper are active participants in and accepted responsibility for the FuSE System Security project. Vision 2025 has this to say about system security in the future:

> "Systems engineering routinely incorporates requirements to enhance systems and information security and resiliency to cyber threats early and is able to verify the cyber defense capabilities over the full system life cycle, based on an increasing body of strategies, tools and methods. Cyber security is a fundamental system attribute that systems engineers understand and incorporate into designs" (INCOSE 2025).

A collaborative team was formed with representation from INCOSE's Systems Security Engineering Working Group, the Systems Engineering Research Center (SERC), the National Defense Industrial Association (NDIA), and the International Society for System Sciences (ISSS). A series

of bi-weekly workshops first deliberated on appropriate strategic foundation concepts for near-term consideration in systems engineering, next key concept factors for each were outlined, and then the work of concept development began with socialization and open recruitment beyond the initial project team.

Context for this paper:

- **Purpose:** Review the initial FuSE security project results and show how systems engineering can improve and advance system security effectiveness in the near term.
- **Problem:** Systems engineering is constraining and impeding rather than enabling and facilitating innovative security.
- **Need:** Strategies for actionable concepts and barrier removal.
- **Intent**: Provide foundation concept descriptions sufficient to inspire and instigate individual concept development and implementation action in the broad-based systems engineering community.

Figure 1 shows the charter that guided the team's work during calendar 2020, evolving as understandings matured. On the left side are three time frames of interest: eventual, near term, and immediate. The initial focus in 2020 was to identify a reasonable and actionable list of foundation concepts that would support achievement of the near-term and eventual objectives.

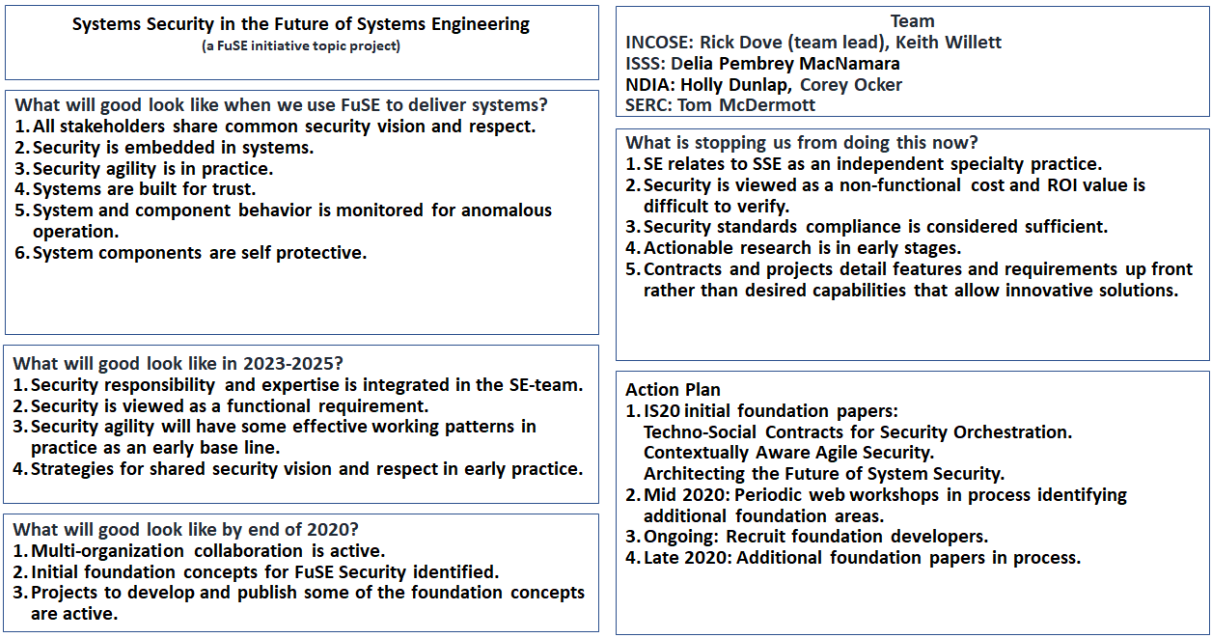


Figure 1. FuSE System Security Project Charter for 2020

Relative to security, "what good will look like" when we use FuSE to deliver systems includes six points evolved as initial objectives.

1. All stakeholders **share common security vision and respect**. Many types of stakeholders are involved in the development, usage, and sustainment of a system designed for purpose. That purpose can be compromised by the weakest security link among the stakeholders, which may stem from insufficient security respect or unresolved priority conflicts.

2. **Security is embedded in systems**. Rather than two engineering groups designing two systems, one intended to protect the other, systems engineering specifies and designs a single system with security embedded in the system and its components.

3. **Security agility** is in practice. The attack community is agile in method innovation and target selection. System security needs a response capability equally agile, architected for proactive composability and reactive resilience.

4. **Systems are built for trust**. Trust is accepted dependence on the system, by both stakeholders and other systems. The reasons for trusting a system need to be built in and evident to all stakeholders.

5. System and component **behaviors are monitored** for anomalous operation. Adversaries innovate new attack methods to evade known-pattern detection screening. System and component behavior outside of normal expectations is a method-agnostic telltale.

6. System **components are self protective**. System componentry is augmented, upgraded, and replaced over time by methods and personnel that cannot be unequivocally trusted.

Eleven foundation concepts that support achievement of the objectives are discussed in the following section. Each concept description attempts to inspire and instigate thinking and involvement in concept development and employment. The concept descriptions outline the nature of the problem, why it is necessary to address the problem, and suggest some notional examples that might inspire solution thinking. The concepts are independent but with synergy such that each may stand on its own but becomes better by virtue of benefits from the others.

Systems engineering is practiced in one form or another in many domains. These foundation concepts are domain agnostic, and some of them may have an early foothold in some domains.

Figure 2 links the foundation concepts to the objectives in a strategic activity web. Linkage lines have no arrowheads as the relationships are synergistic – objectives give purpose to concepts and concepts gives means to objective accomplishments. Foundation concepts are meant to be implementable without dependencies; but a specific concept implementation may provide synergistic relationships with other concepts. Michael Porter makes a point that robustness results from more links in a strategic activity web (Porter 1996). The principle purpose for our display of linkage is to show where the foundation concepts are intended to help achieve FuSE Security long-term objectives. As concepts are developed and implemented additional links are possible.

Figure 2 is not intended to depict a comprehensive security system nor a security architecture; but rather a set of foundation concepts for security improvement appropriate for the current time.

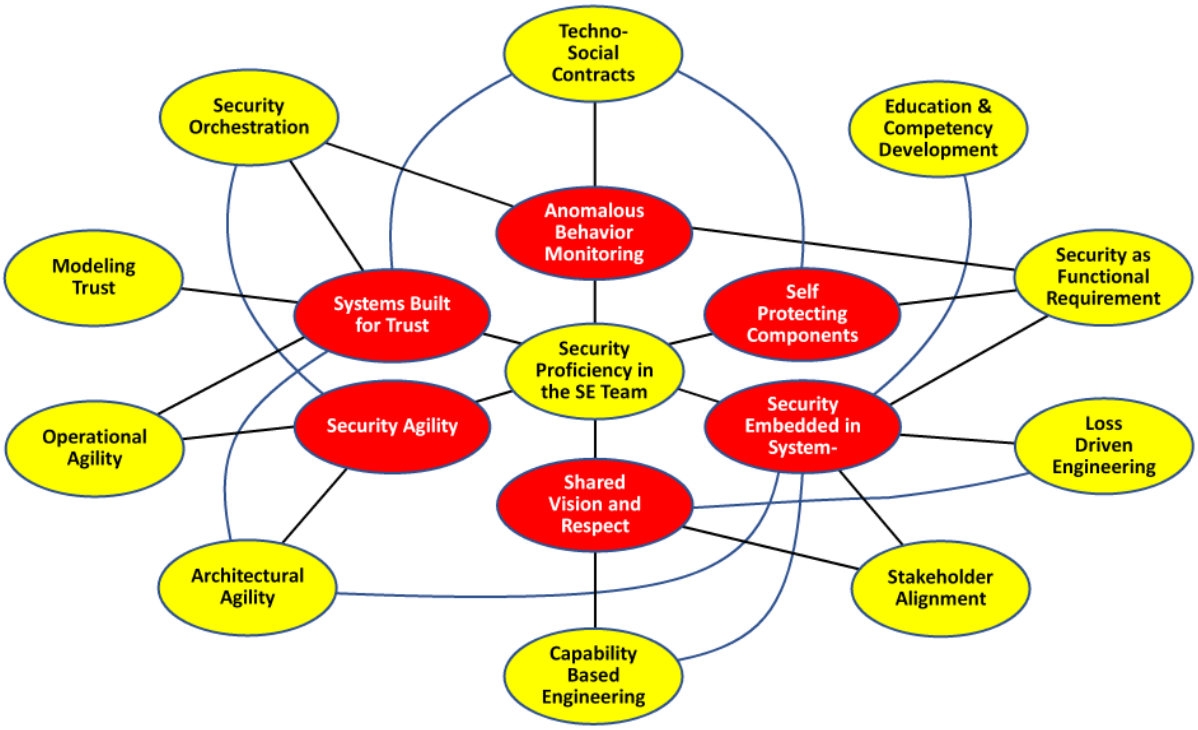


Figure 2. Synergistic linkage of Foundation Concepts (yellow/light ovals with black text) to Objectives (red/dark ovals with white text).

Foundation Concept Descriptions

Criteria for foundation concepts were established as follows:

- Concept has relevance to systems engineering considerations.
- Concept can provide new and useful value to the state of practice.
- Concept value proposition can be articulated in systems engineering terms.
- Concept has notional support in a referenceable knowledge base.
- Concept does not yet have sufficient published exposure for broad-based actionable systems engineering consideration.
- Concept could be implemented now.
- Concept is principally about what to do and why (strategic intent), rather than how (prescriptive tactics), though notional examples of how can augment understanding.

This section provides one-page descriptions for each of the eleven foundation concepts. The focus is on strategic intent, leaving ample room for various approaches. These descriptions generally follow a grouping order of people, process, and technology. The metrics row in the synopsis tables suggests general methods for measuring concept-employment success, with the expectation that concept development for a specific context can be more specific. The notions row provides relevant ideas for inspiring thought without intending to constrain a solution path.

Security Proficiency in the Systems Engineering Team

Table 1. Synopsis: Security Proficiency in the Systems Engineering Team

Problem	Insufficient knowledge of system security engineering at the systems engineering level; communication across knowledge and expertise boundaries.
Need	System security and its evolution effectively enabled by systems engineering activity.
Intent	Integrate socially-sensitive system-level security expertise in the SE team; specify roles and responsibilities across the SE team.
Value	Security sensitive and knowledgeable systems engineering.
Metrics	Security engineering SE-level competencies present; evidence of effective competency application; evidence of accepted roles and responsibilities across the team.
Notions	(Gelosh 2014); (Nejib, Beyer & Yakabovicz 2017).

The professional side of the system adversary community is highly skilled, innovative, and relentless. Targeted systems cannot prevail with fixed defenses against a determined and intelligent attacker. This produces a need for an intelligent defense, one that is highly sensitive to adversarial actions, capable of rapid innovative countermeasures, and equally relentless. All of which is constrained or enabled by early systems engineering decisions that establish system requirements, architecture, and design strategy.

Vision 2025 sees system security as a "fundamental system attribute that systems engineers understand and incorporate into designs." Guidance in this direction can be found in (Nejib, Beyer & Yakabovicz 2017). Understand and incorporate is a minimal and necessary expectation that falls short of proficiency: "a high degree of competence or skill; expertise.[1]" Proficiency is unlikely to be found in systems engineers that haven't spent considerable career time developing breadth and depth in security.

This argues for installing system security engineering proficiency in the systems engineering (SE) team, with key competencies in system security architecture, strategy, and empathy. Security strategy is a process to analyze vulnerabilities and to select protection features that provide acceptable assurance levels to system stakeholders. Empathy is a social attribute that understands how and why to leverage security acceptance and appreciation by all stakeholders who interact with system security, and to balance usability and risk. One of the roles of security proficient personnel in the SE team is to elevate the understandings of others on the team and promote design and architecture strategies relative to security.

Concept development might explore means for finding and embedding appropriate proficiency in the SE team, the nature of SE team interaction and collaboration on security system engineering, or how appropriate proficiency might address each of the FuSE Security objectives and foundation concepts.

---

[1] Google's English dictionary provided by Oxford Languages, 10/17/2020.

Education and Competency Development

Table 2. Synopsis: Education and Competency Development

Problem	Security education is not well integrated with engineering education, creating a skills gap.
Need	Education at all levels focused on security of cyber-physical systems.
Intent	Grow early stage education, university programs, and professional education integrating engineering and security disciplines.
Value	Security becomes more of an engineering design strategy. Design for security is integrated into core systems engineering skillsets.
Metrics	Number of engineers and system engineers trained in system security methods and practices. Design for security fully addressed in SEBOK.
Notions	(ABET 2019); (McDermott, Horowitz & Nadolski 2017); (McDermott 2019); (NIST 2018).

In recent years there has been a strong focus on information security education for information technology (IT) systems. There has not been a related focus on research, knowledge, and education specifically addressing dependable and secure computing in domains such as infrastructure, industrial control, defense, and emerging commercial autonomous systems and the Internet of Things (IoT).

Systems engineering is not addressing security well enough and is not addressing the specific concerns of security emerging in today's embedded systems and cyber physical systems. Programs have gaps in proficiencies related to the security domain and context knowledge at each level across different functional and lifecycle concerns. There is also a clear gap amongst incoming engineers and their understanding of software security versus those that are seasoned and have great domain experience with little system security awareness. Qualification and certification approaches for security professionals in the engineering domains need developed and trained.

Needs in this concept area evolve over time; but a starting point for current needs is suggested in (McDermott 2019) as:

1. Concepts of secure access control to and use of the system and system resources (domain of system security engineering),
2. Understanding of design attributes that minimize exposure of vulnerabilities to external threats (systems security engineering and dependable computing),
3. Understanding of design patterns to produce effects that protect and preserve system functions or resources (dependable computing),
4. Approaches to monitor, detect and respond to threats and security anomalies (cybersecurity),
5. Understanding of network operations and external security services (information systems),
6. Approaches to maintain system availability under adverse conditions.

Supply is created and sustained by demand. Educators and trainers on the supply side and acquirers and developers on the demand side can both contribute to concept development.

Stakeholder Alignment

Table 3. Synopsis: Stakeholder Alignment

Problem	Misalignment of security vision among stakeholders. Inconsistent appreciation for security among stakeholders.
Need	Common security vision and knowledge among all stakeholders.
Intent	Methods to discover and resolve differing points of view.
Value	Commonly accepted security goals and capabilities satisfactory to all stakeholders.
Metrics	Evidence of common knowledge and alignment in stakeholder decision making.
Notions	(Checkland & Poulter 2010); (Reynolds 2011).

In design and development of a new system and in operation of an existing system different stakeholders can influence and affect system security differently. Friction and conflicts can arise when points of view and decision priorities differ, and when a common vision with shared supporting knowledge is lacking. Security has a cost which competes with other stakeholders' values for the system.

This concept explores approaches to identify stakeholder context/perspective, stakeholder needs, discern risk tolerance, resolve conflicts, and normalize a security posture valued and understood by all stakeholders.

Assuming stakeholders desire the system to provide value delivery, they all have interest to eliminate inhibitions to this desire. Stakeholder alignment attempts to remove friction and conflict in the realization of a system security posture acceptable to all relevant stakeholders. Relevant stakeholders are those that constrain, enable, and interact with the security outcome, and include, but are not limited to:

- Those who establish and enforce security policy and procedure.
- Users and usability affected by security policy and procedures.
- Customers who express desires and willingness to invest.
- Design and development engineers who address security needs.
- Managers who allocate resources.

Each of the above stakeholders differ in security perspective and may not realize how that perspective creates friction or is in conflict with others. This is a social issue, not a technical issue; and calls for a social alignment approach which is the responsibility of the systems engineering team to resolve.

Conceptual approaches will consider methods for initial alignment as well as ongoing sustainment of alignment. Notional concepts for inspiration might include soft systems engineering (Checkland & Poulter 2010) and critical systems thinking (Reynolds 2011).

Loss-Driven Engineering

Table 4. Synopsis: Loss-Driven Engineering

Problem	Traditional vulnerability assessments and risk/consequence models for security, safety, and related 'ilities occur too late in the SE process.
Need	Standard metrics and abstractions relevant to all system lifecycle phases.
Intent	Move security and resilience analyses to much earlier development phases, "loss" provides a more consistent metric than traditional risk/vulnerability metrics focused on design.
Value	Concepts of loss and gain can be described and red-teamed well ahead of design. Will greatly improve security requirements definition.
Metrics	Definition & prioritization of loss as a system quality attribute.
Notions	(McDermott & Fleming 2020); (McEvilley 2018); (Young & Porada 2017).

Security, safety, and resilience (and associated dependability attributes of systems) can be explored in an integrated process focused on concepts of loss. A system's resilience is its ability to avoid loss, withstand disruptions that may result in loss, recover from these disruptions, and adapt to internal and external events that may cause disruption. In this context, systems engineering can use a loss-driven methodology for identifying and evaluating resilience alternatives and balancing the effectiveness and affordability of system design alternatives. In particular, the concepts of loss, loss effect, and associated loss scenarios use common abstractions at all phases and levels of the systems engineering process, from mission engineering to detailed design, and from the concept of operations to verification and validation. Loss-driven engineering provides a consistent set of metrics for evaluation of system performance in all lifecycle stages.

McEvilley 2018 defined a working definition of synergistic safety and security design as "Freedom from those conditions that can cause death, injury, or occupational illness; damage to or loss of equipment or property; damage to the environment; damage or loss of data or information; or damage or loss of capability, function, or process." Loss scenarios should be integrated with assurance claims and resulting safety/security requirements and constraints in the design process.

The SE community must formalize approaches to address the potential for loss and associated effects resulting from developing and employing an engineered system. Loss-driven SE is directed by several specialty engineering areas: safety, security, operational risk, resilience, protection, recovery, reliability, and other system 'ilities. The potential for loss associated with a system is currently addressed independently by these different specialty engineering areas. Systems architecting and specialty engineering practices share many commonalities and synergies around how loss and related effects are addressed through requirements, architecture, design, analytics, modeling, simulation, and verification.

Architectural Agility

Table 5. Synopsis: Architectural Agility

Problem	Innovative threats and attacks, and problematic security evolution. [post publication wording improvement]
Need	Readily composable and re-composable security with feature variants.
Intent	Leverage Product Line Engineering concepts and tools.
Value	Security resilience and composable innovation; coherent security evolution.
Metrics	Effectiveness of cyber-relevant response to threat and attack.
Notions	(Clements 2019); (Dove & Schindel 2019); (INCOSE 2019).

Architectural agility is enabled by the Agile Architecture Pattern (Dove & Schindel 2019) with three principle elements: a roster of response capability types with multiple variations within a type, standardized specifications for capability interconnection, and designated responsibilities for managing and evolving the capabilities and interconnection specifications.

Product line engineering (PLE) employs a classic instance of the Agile Architecture Pattern. Instantiating or changing a product within the product family to fill a specific or changed desire is accomplished by selecting appropriate family-shared assets and configuring them as a product configuration, or replacing/augmenting assets in an already deployed configuration.

Shared assets in PLE parlance are "… the 'soft' artifacts associated with the engineering life cycle of the products. A shared asset can be any artifact representable digitally: requirements, design modules, source code, test cases, BOMs, wiring diagrams, documents, and more. They either compose a product or support the engineering process to create a product" (INCOSE 2019).

This concept views security of an entire system as a PLE product. A product line security architecture with a product-family asset management strategy enables security resilience and composable innovation; and facilitates coherent security evolution.

There are commercially available PLE tools, such as (Clements 2019), that could be employed to structure, manage, and evolve security as a product line.

Some security assets may provide features located within system components, and some may provide separate dedicated features. Some will be cyber only, others will be cyber physical and strictly physical. Regardless, structuring all security assets in a product line architecture does not require that the system it is meant to protect be a product line engineered system. What is required is that security assets of similar type with variations have similar interconnect standards, allowing them to replace each other when needed or desired.

This is a promising approach for the architectural underpinnings of the broader concept of agile security; but doesn't provide the operational aspects of agile security, a separate related concept.

Operational Agility

Table 6. Synopsis: Operational Agility

Problem	Effectiveness of detection, response, and recovery. [post publication wording improvement]
Need	Ability for cyber-relevant response to attack and potential threat; resilience in security system.
Intent	Agile architecture, strategy, and operational orchestration.
Value	Innovative and cyber-relevant response capability.
Metrics	Effectiveness of incident response, relevant anticipatory evolution of security system capabilities.
Notions	(Dove & Schindel 2019); (Dove & Willett 2020a); (Maguire 2020).

The purpose of agile security is to provide effective response in an operational environment that can present innovative threats continuously. Architectural agility provides coherent response options for operational agility to select and execute as appropriate to the moment. Effective response requires that events and trends requiring a response are sensed and responded to in a timely fashion, and that both knowledge of the environment and response capabilities evolve with the environment.

(Dove & Willett 2020a) deals with general needs and strategies for agile security in the Future of Systems Security; providing a characterization of the systems security environment and general strategies for dealing with that environment.

Operational agility has three categories encompassing nine general principles (Dove & Schindel 2019):

Sensing:
- External awareness (proactive alertness).
- Internal awareness (proactive alertness).
- Sense making (risk analysis, trade space analysis).

Responding:
- Decision making (timely, informed).
- Action making (invoke/configure process activity to address the situation).
- Action evaluation (verification and validation).

Evolving:
- Experimentation (variations on process ConOps).
- Evaluation (internal and external judgement).
- Memory (evolving culture, response capabilities, and process ConOps).

These principles can offer guidance to the development of operational concept strategies for sensing, responding, and evolving, As additional food for thought, a research-based case study offers novel thoughts and values for considering a self-organizing incident response approach (Maguire 2020).

Capability-Based Security Engineering

Table 7. Synopsis: Capability-Based Security Engineering

Problem	Security often starts with available solutions rather than desired results.
Need	Top-down approach to security starting with desired results/value.
Intent	Standard set of security capabilities to guide architectural conversations.
Value	Clear articulation of desired results; better solution alignment; encouragement for solution innovation.
Metrics	Contract wording and project planning documents that specify required capabilities rather than features.
Notions	(Swarz & DeRosa 2006); (Webb 2006); (Willett 2015).

A capability is an expression of a desired result agnostic of the solution that produces that result. Capability-based engineering (CBE) produces a design based on a set of capabilities and their interactions with which to identify functional needs and substitute any number of solutions that produce those desired results.

This concept will explore and develop a standard set of security capabilities to guide architectural conversations toward a better articulation of desired results and subsequent solution alignment.

If available technology is unable to produce the desired results, the design captures the functional gap and drives innovation. Such knowledge provides for smarter decisions on whether or not to proceed with a system that is less than desired or defer system production until available solutions are more mature.

Solutions may be arranged as capability modules to engage challenges and opportunities more dynamically. Central to the capability perspective is the focus on deploying resources to achieve particular effects or outcomes. Capability based planning is a process to manage evolution as an interrelated set of capabilities versus a set of systems, programs, or solutions. Planning focuses on desired results rather than currently in-use technical products and services.

CBE provides a disciplined approach for purposeful planning, design, building, operation, and evolution based on desired results. CBE take a holistic view of the enterprise and defines, elaborates, and integrates capabilities based on the big picture (Swarz & DeRosa 2006).

Security as a Functional Requirement

Table 8. Synopsis: Security as a Functional Requirement

Problem	As a non-functional requirement, systems security does not get prime system engineering attention.
Need	Systems engineering responsibility for the security of systems.
Intent	Establish security as a functional requirement; inherently raise importance.
Value	Integrates a security view throughout the system engineering lifecycle processes.
Metrics	Presence of effective functional security requirements.
Notions	(ISO/IEC 2017); (OSA n.d.).

Security is an infinite game. Every advance in safeguards meets an advance in adversary exploits. That which constitutes being secure (desired security posture) is driven by continual changes in stakeholder needs, risk tolerance, ecosystem conditions, value-chain, supply-chain, and vulnerabilities. Sustaining a level of acceptable security is ongoing throughout every phase of the system lifecycle including operations from both outside the system (stakeholders, architects, engineers, operators, users/beneficiaries) and within the system; i.e., agile security as a system function.

System security involves selective insertion of additional system functions to monitor, detect, and counter inappropriate or incorrect system behaviors caused by intentional or unintentional disruptions to system operation. (Note: these system functions can also address integrity and safety concerns.) System security engineering is concerned with addressing loss, hazard, and risk in the system but must also be involved in the selection and design of security functions. In order to ensure security is designed in, these must be specified as functional requirements.

A functional requirement is a qualitative description of an *activity to perform* or *purpose to achieve*. An 'activity to perform' is some *action* and a 'purpose to achieve' is some desired *result*. Security has functions and purposes to ensure the system sustains value-delivery under adverse conditions. Without security, the system's extended viability and relevance are left to chance in a nominal world and open to malicious attack in an adverse world.

This concept will explore standard language and standard approach to elicit stakeholder needs and develop system requirements that integrate security within the system.

Modeling Trust

Table 9. Synopsis: Modeling Trust

Problem	Systems Security has moved away from its traditional focus on trust to a more singular focus on risk.
Need	Reinvigorate formal modeling of system trust as a core aspect of system security engineering; address issues of scale with model-based tools and automation.
Intent	Proving a level of system security through evidence based assurance.
Value	Consistent way of evaluating security risk of product or service performance in the environment in which it is expected to operate.
Metrics	The presence and comprehensiveness of models that depict and reflect dependable security functionality.
Notions	(Bell & LaPadula 1973); (Fisher, Launchbury & Richards 2017).

Formal security engineering originated with concepts of trust and trusted systems using formal models. In the early 80s, the U.S. government created the Trusted Computer System Evaluation Criteria (otherwise known as the Orange Book) which integrated the concepts of security functions and security assurance into a single process. This process proved too expensive for concepts of modern computing, and security functions and assurance processes became separated under the Common Criteria (ISO/IEC 15408) standards released in the early 2000s. Since then concepts of risk and assurance have dominated the community and systems were allowed to be deployed without security functions (which became an add-on market). The community needs a return to "Security as a functional requirement" supported by formal security models that support system engineering decisions and V&V activities.

Trust can be defined as accepted dependence of one system on another. Trust models are used to provide insight into the states of dependent systems – to create supporting evidence that a system cannot enter a state where confidentiality, integrity, and availability (CIA) of the system cannot be guaranteed. State-based models suffer from issues of scalability. The system engineering community needs to support research activities and associated methods, processes and tools that address formal models of trust at current and future systems-of-systems scales. Emerging system engineering capabilities such as digital and model-based engineering will provide enabling capabilities. Future technologies employing automation are required for cost effective implementation.

Security Orchestration

Table 10. Synopsis: Security Orchestration

Problem	Disparate security solutions operate independently with little to no coordination.
Need	Tightly coupled coordinated system defense in cyber-relevant time.
Intent	Elaborate *command* portion of orchestration (command and control).
Value	Fast, relevant system defense to sustain value-delivery under adverse conditions.
Metrics	Increase in autonomous system defense. Less people in-the-loop. Cybersecurity within cyber-relevant time.
Notions	(Dove & Willett 2020b); (Iyer 2019); (Maguire 2020).

For [semi-]autonomous systems requiring security decisions in cyber-relevant time, this concept of *Security Orchestration* provides a foundation from which to explore and develop autonomous governance and adjudication logic and rules for dynamic security decisions in operations resulting in fast, relevant, and adaptable system defense. Governance logic addresses strategic goals (interests over time) and tactical objectives (immediate interests); it is the logic behind control. Adjudication logic addresses tensions between strategic and tactical; and, among contending domains like viability, relevance, performance, safety, security, resilience, survivability, and sustainability. Successful resolution of tension is context dependent; i.e., the same stimulus may result in a different response within varying contexts.

Orchestration consists of *command* and *control* features and functions that includes security of the system of interest (SoI) during operations. *Control* is a messaging infrastructure and message set to direct constituent parts of the SoI. The messages are simple commands like START, STOP, BLOCK, and ALLOW among others. The project OpenC2[2] addresses the control feature and is going through the OASIS international standard process . *Command* is the governance and adjudication logic and rule set to make decisions in context of the SoI operating environment. Command is predominantly manual with automation consisting of simple rule sets or logic gates. As artificial intelligence (AI) becomes increasingly viable, we may engineer systems with greater sophistication in what to *observe* (monitor), how to *orient* (understand in context), and how to *decide* (identify and select among viable options). Command then uses control to actuate activity within the SoI (*act*).

Orchestration consists of invoking static solutions in the form of redundancy or standard sequence of events (playbooks); and, dynamic solutions in the form composition (invoking modules) or dynamic development (real-time production of new solution or solution variations; code generation). Encoding of orchestration will draw upon multiple mathematical and technology disciplines working in harmony.

[2] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=openc2

Techno-Social Contracts

Table 11. Techno-Social Contracts

Problem	Insufficient detection capability for innovative attack methods with dedicated purpose security components. [post publication wording improvement]
Need	Quick detection and mitigation of known and unknown attacks.
Intent	Autonomous collaboration for mutual/shared protection.
Value	Widely distributed autonomous behavior-based security.
Metrics	Threats detected and mitigated with this method undetected by other in-place methods.
Notions	(Dove & Willett 2020b); (DHS 2011); (Duffy 2004); (Rose et al. 2020); (Rousseau 1762).

On the Social Contract, a book by philosopher Jean-Jacque Rousseau (1762), explains that people banded together in communities for the purpose of mutual protection. A social contract is an implicit cultural agreement among members of a society that "essentially binds the members into a community that exists for mutual preservation."

The techno-social contract concept views systems as communities of technical components, where security of the community and its technical members can benefit from collective, distributed, interactive behaviors for purposes of mutual protection. The general concept was introduced in (Dove and Willett 2020b) with references to prior work by others and component-embedded functional strategies for: self protection, self awareness, self behavior judgement, self behavior mitigation, peer behavior judgement, peer behavior mitigation, peer collaboration, adaptable attention priorities, diversity, and heterogeneous awareness.

This concept's thesis is that mutual protection behavior among technical system components is both beneficial and possible. Beneficial in that collaboration, cooperation, and teaming among system elements during system operation offers novel strategy for quick detection and mitigation of innovative security threats. Possible in that human and animal communities employ effectively demonstrated approaches, and some work in non-human socially behaving system aggregations already exists (Duffy 2004).

Concept development needs to address means for enabling, designing, and implementing techno-social contracts. Enabling includes rethinking security strategy and systems architecture. Designing includes the selection of techno-social strategies and the development of tactical concepts, and establishing the human role in community governance vs. autonomous governance, higher authority decision making, and contract enforcement. Implementing includes the employment of an agile systems engineering life-cycle approach that facilitates continuous learning and evolution.

Conclusion

The FUSE initiative has recognized that changes are needed in various elements of systems engineering and the proficiencies of a systems engineer to address current and future system security challenges. This study links eleven foundation concepts to six objectives in the future of systems security. We would note that all of these are current needs, as the systems engineering community is just one domain trying to catch up to today's security issues. Systems engineering can take a leadership role in the future of system security by ensuring that these foundation concepts are "designed in" to future systems and development teams: "cyber security is a fundamental system attribute that systems engineers understand and incorporate into designs" (INCOSE 2014).

The concept descriptions emphasize that security engineering must be fundamental to systems engineering, not just a specialty discipline. Security concepts must be fundamental to engineering education, and security proficiency must be fundamental in development teams. Security functions must be foundational across product lines and incorporation must be as agile as the threat has demonstrated. Security fundamentals must be clearly understood by stakeholders and effectively evaluated in a way that considers broad goals with security functions and outcomes. Many of these concepts are not new but they have not adapted well to systems with increasing complexity and connectivity and sociotechnical impact.

We invite the systems engineering community to comment on these and suggest new foundation concepts. We invite the systems engineering research community to address these concepts with new methods, processes, and tools that ensure future systems are secure.

This project's work in 2020 focused on identifying a set of foundational concepts upon which improved security in the immediate future of systems engineering can be built. Work in 2021 and beyond moves the project team's focus to concept development, implementation, and practice. Membership in the core team will evolve to meet this focus. General team objectives remain the same: inspire and instigate – concept development, implementation, and practice requires community initiative and involvement.

Contact project leadership if you would like to be a part of this activity in any way: concept strategy development, implemented case study exposure, experimental implementation, community instigation and inspiration, or something else you feel would be useful.

References

ABET 2019, Criteria for Accrediting Engineering Programs, 2019 – 2020 p. 19: Program criteria for cybersecurity engineering and similarly named engineering programs, <www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020>.

Bell, DE & LaPadula. LJ 1973, 'Secure computer systems: mathematical foundations', MITRE Corporation, <http://www-personal.umich.edu/~cja/LPS12b/refs/belllapadula1.pdf>.

Checkland, P & Poulter, J 2010. 'Soft Systems Methodology', Reynolds, M & Holwell, S eds, in: *Systems Approaches to Managing Change: A Practical Guide*, London: Springer, pp. 191–242. <www.kenniscentrumtoerisme.nl/images/archive/2/25/20190930164023%21Systems-Approaches-to-Managing-Change.pdf#page=198>

Clements, PC 2019, 'Product line engineering comes to the industrial mainstream', INSIGHT vol. 2, no. 2, International Council on Systems Engineering, August 2019.

DHS 2011, 'Enabling distributed security in cyberspace – Building a healthy and resilient cyber ecosystem with automated collective action', March 23, 2011, <www.dhs.gov/xlibrary/assets/nppd-cyber-ecosystem-white-paper-03-23-2011.pdf>

Dove, R & Willett, KD 2020a, 'Contextually aware agile security in the future of systems engineering', International Council on Systems Engineering, IS20 virtual conference conducted on Cape Town, South Africa time, 20-22 July 2020, <www.parshift.com/s/200718IS20-FuSEAgileSecurity.pdf>.

———2020b, 'Techno-social contracts for security orchestration in the future of systems engineering', International Council on Systems Engineering, IS20 virtual conference conducted on Cape Town, South Africa time, 20-22 July 2020, <www.parshift.com/s/200718IS20-FuSETechnoSocialContracts.pdf>.

Dove, R & LaBarge, R 2014, 'Fundamentals of agile systems engineering – Part 1', International Council on Systems Engineering IS14, Las Vegas, NV, 30 June – 3 July 2014, <www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1.pdf>.

Dove, R & Schindel, W 2017, 'Case study: Agile SE process for centralized SoS sustainment at Northrop Grumman', International Council on Systems Engineering IS17, Adelaide, Australia, 17-20 July 2017, <www.parshift.com/s/ASELCM-03NGC.pdf>.

———2019, 'Agile systems engineering life cycle model for mixed discipline engineering', International Council on Systems Engineering IS19, Orlando, FL, USA, 20-25 July, <www.parshift.com/s/ASELCM-05Findings.pdf>.

Duffy, B 2004, 'Robots social embodiment in autonomous mobile robotics,' International Journal of Advanced Robotic Systems, vol. 1, no. 3, pp. 155-170, 2004, <https://journals.sagepub.com/doi/pdf/10.5772/5632>.

Fisher, K, Launchbury, J & Richards, R 2017, 'The HACMS program: using formal methods to eliminate exploitable bugs', Phil. Trans. R. Soc. A. 375:20150401, 13 October 2017, <https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0401>.

Gelosh, D 2014, 'Systems security engineering competency model', NDIA SE Conference. Springfield, VA, 29 October 2014, <https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2014/system/17016WedTrack1Gelosh.pdf>.

INCOSE 2014, *A World in Motion – Systems Engineering Vision 2025*, International Council on Systems Engineering, July 2014, <https://www.researchgate.net/publication/277019221_A_World_in_Motion_-_Systems_Engineering_Vision_2025>.

INCOSE n.d. 'The Future of Systems Engineering', <www.incose.org/about-systems-engineering/fuse>.

INCOSE 2019, 'Feature-based systems and software product line engineering: a primer', INCOSE TP-2019-002-03-04-04, prepared by the PLE International Working Group. International Council on Systems Engineering. San Diego, CA, USA.

ISO/IEC 2017, ISO/IEC 15408: Common Criteria for Information Technology Security Evaluation, version 3.1, revision 5, April. <www.commoncriteriaportal.org/cc/>

Iyer, A 2019, *Security Orchestration for Dummies*, John Wiley & Sons. <https://virtualizationandstorage.files.wordpress.com/2019/04/security-orchestration-for-dummies-demisto-special-edition.pdf>.

Maguire, L 2020, 'How many is too much? Exploring costs of coordination during outages', proceedings, video and transcript, QCon, London, 11 May 2020, <www.infoq.com/presentations/incident-command-system>.

McDermott, T 2019, 'Emerging education challenges for resilient cyber physical systems', 29th Annual INCOSE International Symposium, Orlando FL, 20-25 July 2019.

McDermott, T, Horowitz, B & Nadolski, M 2017, 'Human capital development – resilient cyber physical systems, Technical Report SERC-2017-TR-113, Systems Engineering Research Center, 29 September 2017, <https://apps.dtic.mil/dtic/tr/fulltext/u2/1040186.pdf>.

McDermott, T & Fleming, C 2020, 'Methods to evaluate cost/technical risk and opportunity decisions for security assurance in design, Technical Report SERC-TR-2020-005, Systems Engineering Research Center, 12 June 2020, <https://web.sercuarc.org/documents/technical_reports/1596820233-A013_SERC%20ART%20004_Technical%20Report%20SERC-2020-TR-005.pdf>.

McEvilley, R 2018, 'Leveraging system safety to improve system security', 21st Annual National Defense Industries Association (NDIA) Systems and Mission Engineering Conference, 22-25 October, Tampa, FL, <https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2018/systems/Thurs_21412_McEvilley.pdf>.

Nejib, P, Beyer, D & Yakabovicz, E 2017, 'Systems security engineering: what every system engineer needs to know', International Council on Systems Engineering IS17, Adelaide, Australia, 15-20 July 2017.

NIST 2017, NIST Special Publication 800-181, National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework, National Institute of Standards and Technology, August 2017, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-181.pdf>.

OSA. n.d. Open Security Architecture, <www.opensecurityarchitecture.net/cms>.

Porter, ME 1996, 'What is strategy?', Harvard Business Review, November-December, 1996.

Reynolds, M 2011, 'Critical thinking and systems thinking: towards a critical literacy for systems thinking in practice, Chapter 2 pp. 37-68 in: *Critical thinking*, Editors: Horvath CP & Forte, JM, Nova Science Publishers, Inc. <www.researchgate.net/publication/287571708_Critical_thinking_and_systems_thinking_Towards_a_critical_literacy_for_systems_thinking_in_practice>.

Rose, S, Borchert, O, Mitchell, S & Connelly, S 2020. Zero Trust Architecture. Draft (2nd) NIST Special Publication 800-207, February. 2020, <https://doi.org/10.6028/NIST.SP.800-207-draft2>.

Rousseau, J-J 1762, *On the Social Contract*, English translation by Maurice Cranston, Penguin Publishing Group, 28 June 1968, Barnes & Noble key points synopsis and full text at <www.sparknotes.com/philosophy/socialcontract/characters>.

Swarz, RS & DeRosa, JK 2006, 'A framework for enterprise systems engineering processes' International Conference on Software and Systems Engineering, <www.mitre.org/sites/default/files/pdf/06_1163.pdf>.

Webb, M. 2006, 'Capabilities-based engineering analysis (CBEA)', International Conference on Complex Systems, NECSI, Boston, MA, 29 June 2006, <www.researchgate.net/publication/228596963_Capabilities-Based_Engineering_Analysis_CBEA>.

Willett, KD 2015, 'Capability-based engineering approach to integrated adaptive cyberspace defense (IACD), IAD Information Assurance Symposium, Washington D.C., July 2015, <www.researchgate.net/publication/346012308_Capability-Based_Engineering_Approach_to_Integrated_Adaptive_Cyberspace_Defense_IACD>.

Young, W & Porada, R 2017, 'System-theoretic process analysis for security (STPA-SEC): Cyber security and STPA', 2017 STAMP Conference, Boston, MA, 27 March 2017, <https://psas.scripts.mit.edu/home/wp-content/uploads/2017/04/STAMP_2017_STPA_SEC_TUTORIAL_as-presented.pdf>.

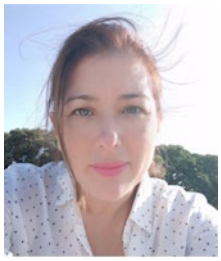
Biography

Rick Dove is CEO of Paradigm Shift International, specializing in agile systems and security research, engineering, and project management; and an adjunct professor at Stevens Institute of Technology teaching graduate courses in agile and self-organizing systems. He chairs the INCOSE working groups for Agile Systems and Systems Engineering, and for Systems Security Engineering. He is an INCOSE Fellow, and author of *Response Ability, the Language, Structure, and Culture of the Agile Enterprise*.

Dr. Keith D. Willett is a senior strategist and enterprise security architect for the United States Department of Defense**.** He has a PhD in systems engineering from Stevens Institute of Technology. He is co-chair for the INCOSE working groups on Systems Security Engineering and Agile Systems & Agile SE. Dr. Willett is sole author of *Information Assurance Architecture* and coauthor of three books including *Multisystemic Resilience – Adaptation and Transformation in Contexts of Change* (Oxford Press) February 2021.

Tom McDermott is the Deputy Director and Chief Technology Officer of the Systems Engineering Research Center at Stevens Institute of Technology in Hoboken, NJ. He leads research on Digital Engineering transformation, education, security, and artificial intelligence applications. Mr. McDermott also teaches system architecture concepts, systems thinking and decision making, and engineering leadership for universities, government, and industry. He serves on the INCOSE Board of Directors as Director of Strategic Integration.

Holly Dunlap is a Senior Principal Engineer at Raytheon Technologies with over 20 years experience. Ms. Dunlap is responsible for a holistic approach to program protection to manage and balance the security-relevant risks within platforms or embedded defense systems. Ms. Dunlap is currently the chair of the National Defense Industrial Association (NDIA) Systems Engineering Division and has chaired the NDIA Systems Security Engineering (SSE) Committee for 9 years.



Delia Pembrey MacNamara is current President of the International Society for the Systems Sciences and chairs the Science, Spirituality and Systems Science Special Integration Group. She is on the Board of Trustees for the American Society for Cybernetics and is a PhD candidate at the University of Hull in the UK researching the systems concept of 'Boundary', applying general systems theory, critical systems thinking and cybernetics. Delia works for the Australian Government in the artificial intelligence, automation and ethics domain.

Cory Ocker is a Secure Systems Engineering Manager at Raytheon Technologies focused on securing embedded systems. He previously served as a government civilian securing multiple systems including the F-15, AIM-260, and various other weapon programs as an Agent of the Security Control Assessor. He chairs the NDIA Systems Security Engineering (SSE) Committee where he works with the Office of the Secretary of Defense and other services on establishing policy, practices, and guidance for the SSE community.