Methods to Evaluate Cost/Technical Risk and Opportunity Decisions for Security Assurance in Design

Technical Report SERC-TR-2020-005

12 June 2020

Principal Investigator: Tom McDermott, Stevens Institute of Technology

Co-Principal Investigator: Cody Fleming, University of Virginia

Research Team:

Stevens Institute of Technology: Megan M. Clifford

University of Virginia: Tim Sherburne

Georgios Bakirtzis, Krista Rand

Sponsor: Office of the Under Secretary of Defense for Research and Engineering (USD(R&E))

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract [W15QKN-18-D-0040/0001].

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense nor ASD(R&E).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full Description
SERC	Systems Engineering Research Center
AADL	Architecture Analysis and Design Language
APT	Advanced Persistent Threat
CASE	Cyber Assured Systems Engineering
CPS	Cyber-Physical System
CRWS	Cyber Resilient Weapon Systems
CSES	Cyber Security Econometrics System
CSRM	Cyber-Security Requirements Methodology
DARPA	The Defense Advanced Research Projects Agency

DE and DES	Digital Engineering and Digital Engineering Strategy
EFFBD	Enhanced Functional Flow Block Diagram
HACMS	High-Assurance Cyber Military Systems
ICS	Industrial Control Systems
IEEE	The Institute of Electrical and Electronics Engineers
LAN	Local Area Network
MA	Mission Aware
MAAP	Mission Assurance Analysis Protocol
MBA	Model-Based Assurance
MBE	Model-Based Engineering
MBSA	Model-Based Systems Assurance
MBSE	Model-Based Systems Engineering
ME	Mission Engineering
MPTs	Methods, Processes, and Tools
NDS	National Defense Strategy
NPEC	Nuclear Power Engineering Committee
OGCPS	Oil and Gas Cyber Physical System
OMG	Object Management Group
OUSD	Office of the Under Secretary of Defense
PIG	Pipeline Intervention Gadget
R&E	Research & Engineering
RT	Research Task
SCADA	Supervisory Control and Data Acquisition
STPA	System Theoretic Process Assessment
STPE	Strategic Technology Protection and Exploitation
UCA	Unsafe Control Action

EXECUTIVE SUMMARY

Background and Purpose:

This research addresses needs defined by the Office of the Undersecretary of Defense for Research and Engineering (OUSD/R&E), Strategic Technology Protection and Exploitation (STPE) Division to develop standard approaches to "design in" security and resilience for current and future weapon systems. The proposal closely aligns with the OUSD/R&E's Digital Engineering Strategy (DES) and the Cyber Resilient Weapon Systems (CRWS) initiative. It extends ongoing research in Security Engineering within the Systems Engineering Research Center (SERC) to a broader definition of system assurance. It addresses a gap in current systems engineering methods, processes, and tools (MPTs) associated with early-phase requirements assessment in Cyber Resilience system trades. Our STPE research sponsors are specifically interested in developing new standard approaches that combine security assurance and safety assurance (as well as other assurance concerns) in a common, model-based, systems engineering process. This integrated view is shown in Figure 1 [1].

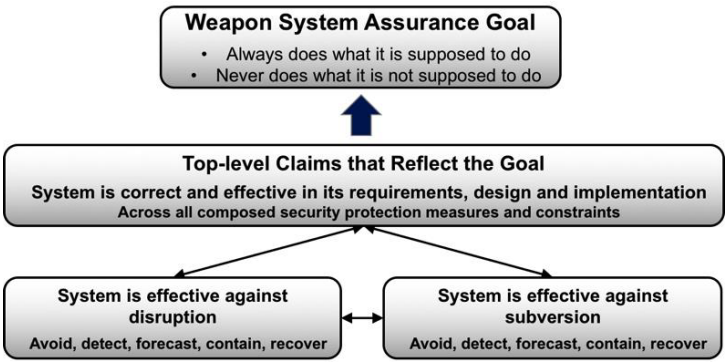

Figure 1. Weapon System Assurance Goals

This research responds explicitly to sponsor desires to leverage relationships between system safety and systems engineering to improve system security and resilience. System safety has a history of successfully integrating practice into the systems engineering process to enable more interdisciplinary collaboration and better-informed trades [1]. Reed and McEvilley define a working definition of synergistic safety and security 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, assurance claims, goals, and resulting safety/security requirements and constraints are used in the combined evaluation of safety and security evidence in the design process. Assurance claims are system attributes evaluated in system engineering trades. The resulting system design must follow methodologies that consider need, design and evaluation rigor, and return on investment.

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 [2]. In this context, system assurance is a loss-driven methodology for identifying and evaluating resilience alternatives and balancing the effectiveness and affordability of system design alternatives. It considers four modeling goals:

1) a model of the system and its mission, operational tasks, behaviors, and structure
2) modeling the concept of maximum reasonable assurance – the decision process that considers system performance safety, security, dependability, and associated characteristics, and determines the appropriate responses to malicious and non-malicious disruption to the system that could result in losses
3) models that capture engineering rigor – the engineering methods and processes that support the specification, architectural definition, design, analysis, and verification & validation of the system, and
4) creation of a system resilience model – a model that communicates the system, the threats to the system (disruptions and resultant losses), the assurance decisions (requirements and constraints), and the countermeasures (design decisions and resilience modes) added to the system model.

The research focused on capturing all four modeling goals in a consistent environment using Model-Based Systems Engineering (MBSE) methods, processes, and tools. A primary outcome of this research is the development and maturation of a meta-model capturing central concepts of a system (operations, function, structure, requirements), assurance (loss, loss effect, and loss scenario), and resilience design (functions that avoid, withstand, recover, and adapt) into MBSE tool constructs. Figure 2 [3] summarizes the process goals for the research.

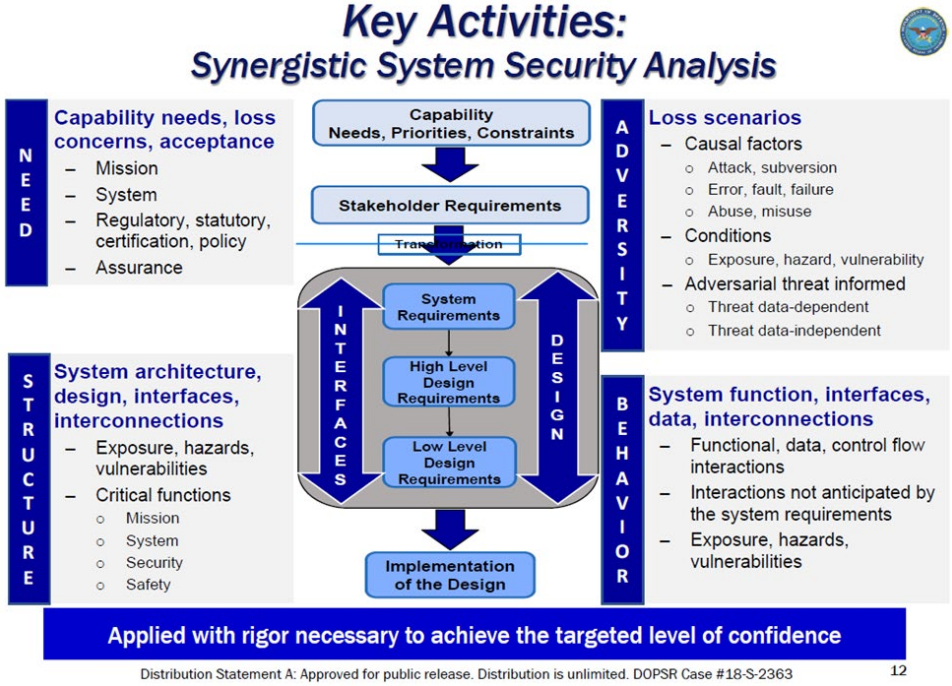

Figure 2. Synergistic Safety and Security Activities in the Systems Engineering Process [3].

For several years, a principal focus of the Trusted Systems research thrust within SERC has been developing methods and tools that support system design for cyber resilience in cyber physical systems. This body of work features the development of the Mission Aware (MA) framework for integration and alignment of cyber engineering requirements with the system development lifecycle and systems engineering processes. MA includes techniques to evaluate cyber physical system threats and attacks, requirements and design concepts for cyber resiliency, and model-based tools for selecting resilient architectures. The MA framework's centerpiece is a risk analysis that integrates the perspectives of mission owners, systems engineers, and adversary red teams into a common model-based form.

MA was developed through a series of SERC research efforts, notably RT-156, RT-172, RT-191, RT-196, WRT-1013, and this effort: ART-004. WRT-1013 developed a meta-model that can be used to derive model-based systems engineering (MBSE) representations of systems [4] [5] [6] [7]. The meta-model includes loss scenarios, hazards, threat activities, system resilience modes of operation, and control-driven representations of security requirements. The meta-model captures the results of a standard Cyber Security Requirements Methodology (CSRM) intended to be conducted through the early stages of system definition and development, which was matured in RT-191 and RT-196. The MA Meta-Model was demonstrated in a current-generation MBSE software suite. ART-004 extends this work to a formal methodology for assurance case reasoning in resilient cyber design that can be standardized across the DoD Mission Engineering and system definition phases of a weapon system.

Research Goals and Results:
The research goals center on two primary questions:

1) Can we define a standard methodology to integrate cyber resilience analysis into systems engineering activities building from the success of safety engineering activities?
2) Can we define a framework for decision metrics that consider both the cyber threat and system model to inform tradespace analysis of the system resilience model?

Figure 3 shows at a high level, the strategy of the project to extend previous SERC MA research to the two central research questions.

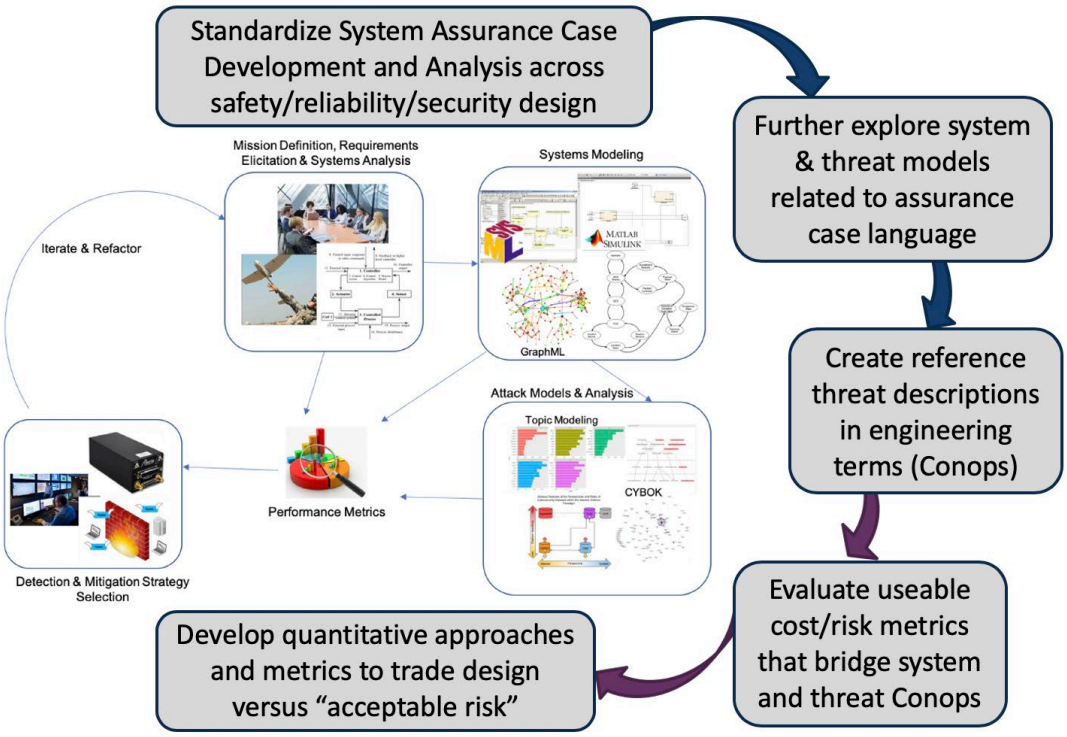

Figure 3. Expand Mission Aware: Rigorous Functional Security Analysis and Modeling Process

The research produced several contributions to the fields of safety, security, and resilience:

1. Development of a candidate approach for "loss-driven systems engineering."
2. Assessment of the application of different assurance standards and modeling methods in consideration of a system's combined safety and security characteristics
3. Exploration of previous research on model-based system assurance and its ability to extend to more complex systems-of-systems.
4. Standard means using a Conops format and models to express concepts of threat, resilience, safety, and assurance from the mission level down to design, and from the operational view to the engineering views.
5. Development of a metrics framework that links together threat motivation with system loss trades that can be expressed as decision metrics at multiple levels of the system.
6. Demonstration of the approach in a publicly accessible modeling case study.

Loss-Driven Systems Engineering

The systems engineering community seeks to formalize an approach to address the potential for loss and associated effects resulting from developing and employing an engineered system. While much of systems engineering focuses on the delivery of desired capabilities, loss-driven systems engineering addresses potential losses associated with the system of interest. Loss-driven systems engineering 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. System attributes such as resilience and infrastructure protection have a common association with these specialty areas through the concept of loss and associated loss impacts. These are shown in Figure 4. 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. 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 [1].

The goal of capturing all of these specialty perspectives in an integrated architecture model using MBSE tools is a crucial outcome of this research.

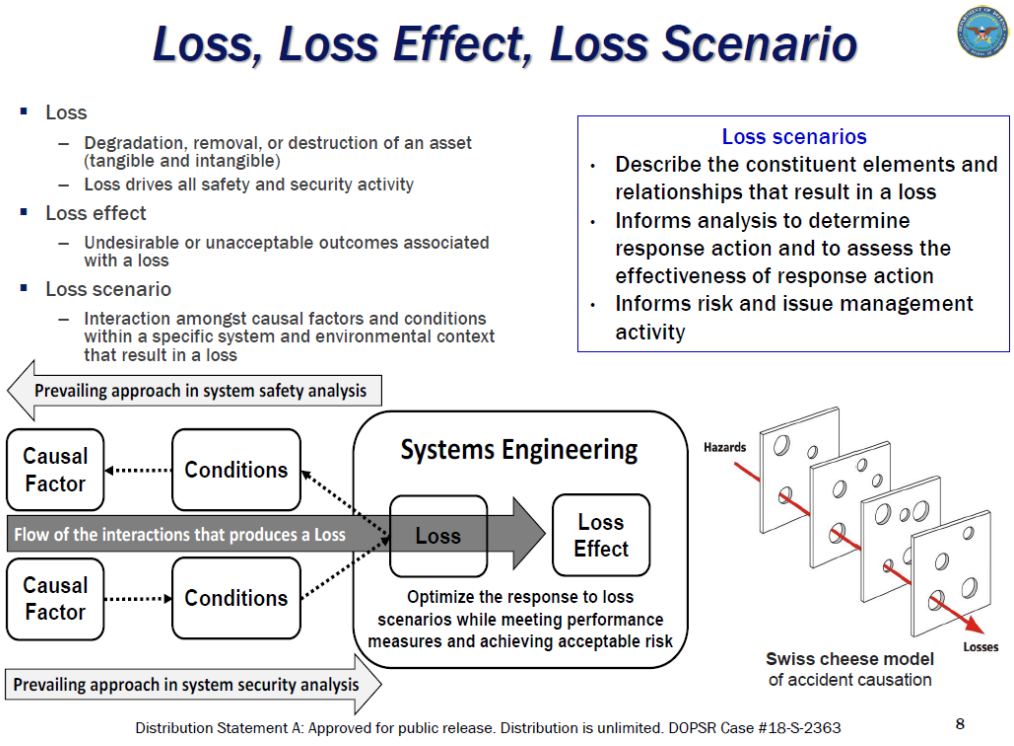

Figure 4. Loss, Loss effect, and Loss Scenario [1].

Integrating assurance standards and modeling methods

Assurance, as defined, is grounds for justified confidence, gained before depending on a system, that a claim about dependability, safety, or security has been (or will be) achieved. A claim is a true-false statement about one of these properties of a system [8]. Assurance is related to the "requirements of a property of a system." As defined by NATO, system assurance is the justified confidence that a system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the

life cycle. System “functions as intended” and “free of exploitable vulnerabilities” represent the system's highest-level properties [9]. The cybersecurity community has focused too firmly on exploitable vulnerabilities. It needs a much more rigorous approach to gain confidence that the system functions as intended in the presence of external threats. This confidence is achieved by system assurance activities, including a planned, systematic set of multi-disciplinary activities to meet the acceptable measures of system assurance and manage the risk of exploitable vulnerabilities. One can argue that “functions as intended,” for systems of any complexity, requires a modeling method that relates system function to the requirements properties of a system that define its dependability, safety, and security. These properties can be constraints on the system function or additional system functions that support assurance activities. The CSRM and MA Meta-model, as defined in this research, provide a standardized approach to link modeling, assurance cases, system constraints (requirements), and what we term “resilience modes” (additional system functions) in an MBSE toolset [6]. The MA Meta-model provides a standard set of design patterns to formalize this approach.

An assurance case, per ISO/IEC/IEEE 15026, is a reasoned, auditable artifact that supports the contention that an assurance claim has been satisfied, including systematic argumentation and supporting evidence. The assurance case components include claims, arguments, evidence, justifications, and assumptions. The goal of an assurance case is to communicate the assurance properties to stakeholders, informing their decision-making, and providing the necessary confidence in the system [10]. This report will show how a well-structured assurance pattern in an MBSE model improves standard assurance artifacts. It describes both the intended function and exploitable vulnerabilities in a common pattern. Neither standalone assurance cases in an argument-based format, nor tables of vulnerabilities, hazards, and risk, can compete with a functional model for communicating the linkage between vulnerabilities and intended function.

Assurance cases capture the subjective argument and structure judgment through claims which must be supported by evidence. Assurance cases include justification based on different methods of reasoning about the system properties. Aspects of dependability, safety, and security differ in their ways of reasoning. These methods produce evidence that can be qualitative or quantitative, deterministic, or non-deterministic [11]. Ideally, the development of an assured system would include cases that move from qualitative to quantitative and non-deterministic to deterministic as the system lifecycle matures. A model is a useful means to capture and manage the relationships between these different reasoning methods. Figure 5 shows the relationship between different assurance reasoning approaches and the system decision lifecycle.

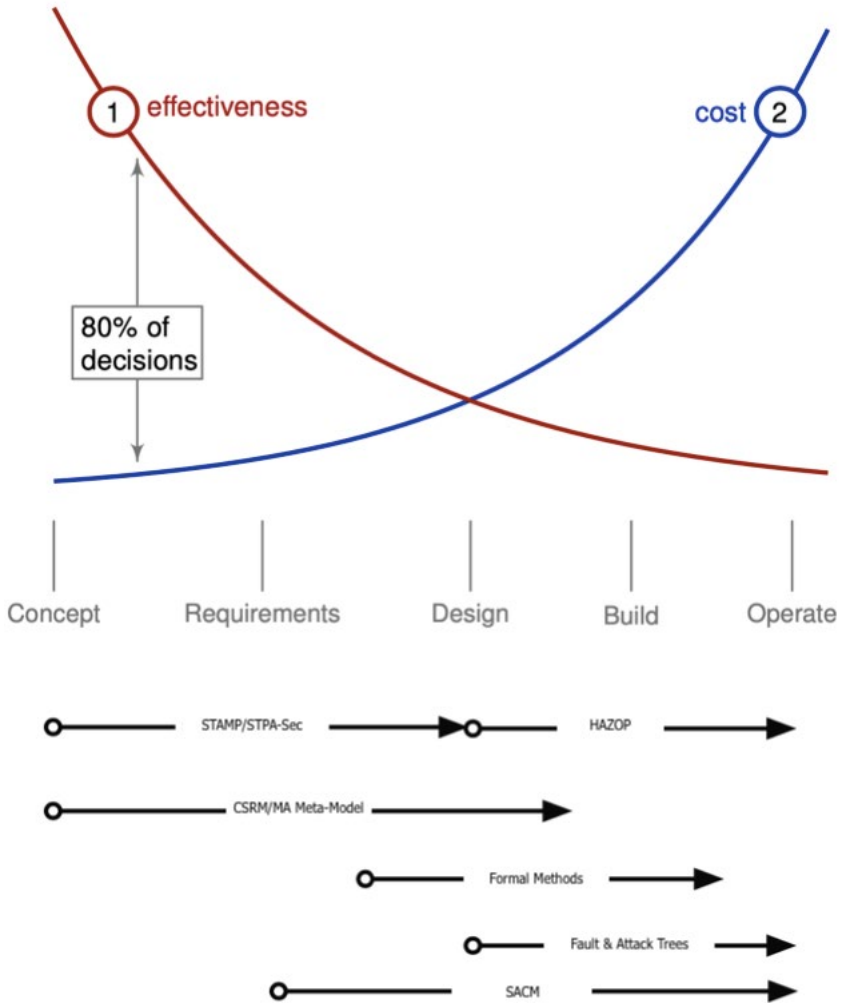

Figure 5. Assurance Methodologies at the Various Stages of the System Lifecycle

The different assurance methods loosely map to varying timelines during the system lifecycle. This project has been particularly interested in reasoning at higher abstraction levels, where the system's intended function would be initially defined. The cyber resilience process is heavily focused on high-level system behaviors and associated mission resilience features of the system, assuming that not all exploitable vulnerabilities can be eliminated. The assurance process should be started in conceptual stages, particularly mission engineering and system definition activities. In these stages, concepts of dependence and loss can be defined and prioritized as requirements, even though system vulnerabilities cannot. The proximity to the stage of the lifecycle addressed, reality, abstraction/level of fidelity, interoperability, tool support, and verification and validation were most closely met by the System Theoretic Accident Methods and Processes (STAMP) and associated System Theoretic Process Assessment (STPA) tools. The unification of the STPA technique for safety analysis and derived STPA-Sec for security analysis has proved to guide in-depth security analysis to the most vulnerable and critical components of a system. This research confirms that STPA, developed in the safety community, is the most effective method for reasoning about security assurance. STPA-Sec has been integrated into the CSRM and MA Meta-

Model. The two of these together support the argument about assurance in a more significant portion of the lifecycle. The Structured Assurance Case Meta-model (SACM) standardizes the structure and use of assurance case language integrates well with CSRM and the MA Meta-Model but uses different language constructs and reasoning approaches. SACM and other argument or claim based approaches do not formalize the concept of architectural design patterns for safety, security, and resilience, so there are limits in describing resilience, for instance, in these approaches. Other approaches such as Hazard and Operability Analysis (HAZOP), fault and attack trees, and formal methods are more useful once the system architecture and preliminary design have been described (once component classes have been selected) [8] [12] [13]. This research suggests CRSM and the MA Metamodel are valuable additions to the assurance tool suite and can be more fully integrated with system architecture models.

A Complex Systems Case Study

An initial step in this research was to select a case study that could be used to apply and demonstrate the methods, processes, and tools used or developed in the project. The case study needed to be an example of a cyber-physical system, be used in a complex system-of-systems, and a case where the relationships between threat goals and benefits could be quantified concerning costs and risks associated with system resilience. We also wanted an openly publishable example. The selected case study represents a unique example of an advanced persistent threat in a critical infrastructure system exploited for monetary gain: pipeline and oil pumping stations and associated pipeline oil delivery operations and market activities. The case study was selected since it (1) represents a plausible APT in a critical infrastructure managed both through the human operator and cyber-physical control systems; (2) the relative scope of the threat team's effort can be estimated, and the monetary gains from the attack can be modeled; and (3) it represents an exploitable gap in existing security practices in large systems tied to multiple organizations in the supply chain. Also, the case study architecture has a scope that can be modeled in present MBSE tools.

Standard means to express concepts of threat, resilience, safety, and assurance in a model

This research firmly established the credibility of the MBSE MA Meta-Model as an effective path toward security assurance in early-stage design. The general approach developed serves as a basis for a repeatable, yet flexible approach. The framework and foundations established in the research are ready for transition. A particular transition focus is toward mission engineering and early-stage system definition in the government MBSE modeling settings. Still, the techniques can and should be applied consistently across all program lifecycle phases. Modeling assurance cases and resulting resilience modes of the system is a crucial aspect of system architecting and the MBSE MA Meta-Model provides a standard architectural representation for loss scenarios, assurance requirements, and resilience features of the architecture. Assurance cases are intended to be developed and maintained for the full lifecycle. The MBSE MA Meta-Model provides a standard approach to capture all aspects of the assurance process.

The case study and model were integrated into a form where researchers analyzing the threat approach to exploit the system worked with researchers developing the Meta-Model. The goal was to simulate the reasoning concerning threats and modeled assurance properties in a realistic

setting. As this process proceeded, the operational relationships between threat and assurance were captured in the form of a standard Concept of Operations. This approach proved useful in the CSRM process used on this project and aligned well with the way the DoD documents early-stage concept definition activities. The CONOPS format was extended to capture the system changes needed to counter cyber threats in an operational context. The CONOPS table of contents are included as Appendix B in this report.

Metrics framework that links together threat motivation with system loss trades

The oil and gas case study allowed definition of resilience metrics for evaluating the effectiveness of resilience solutions in response to safety and security violations while achieving operational priorities. The case study's meta-model relates the expert and operator perspectives, which are required for priority ranking of system losses, likelihood, and severity determination for attack vectors to evaluate the effectiveness and complexity of resilient modes. An essential set of metrics at the full system level includes attacker gain and defender loss, which have been poorly described in other security analysis methods. Other important evaluation metrics for resilient system modes include the operational impact and the time budget for system recovery. Recovery time includes detection time, isolation time, and restore time, including any operator decision time. System simulation can evaluate the recovery ratio for critical system functions under various system loads and simulated attack patterns. Tradespace analysis, based on resilience metrics, enables specifications of a system that responds to safety and security violations while achieving operational priorities within programmatic cost and time constraints. The use case, modeling, and Meta-Model showed the feasibility of evaluating such metrics for a given system. The research supports a resilience evaluation metrics framework that links together threat motivation with system loss trades, but further progress is needed to formalize.

Demonstration of the approach in a publicly accessible modeling case study

The published, open-source GitHub model is decomposed and organized according to the Mission Aware methodology using the Vitech GENESYS MBSE modeling tool, which was extended with our Meta-Model. The particular tool is not necessary to use the Meta-Model. We use the tool and its associated diagrams to visualize the different model views as defined by the Meta-Model. The public model can be explored at:
https://coordinated-systems-lab.github.io/pipeline-cps/index.html

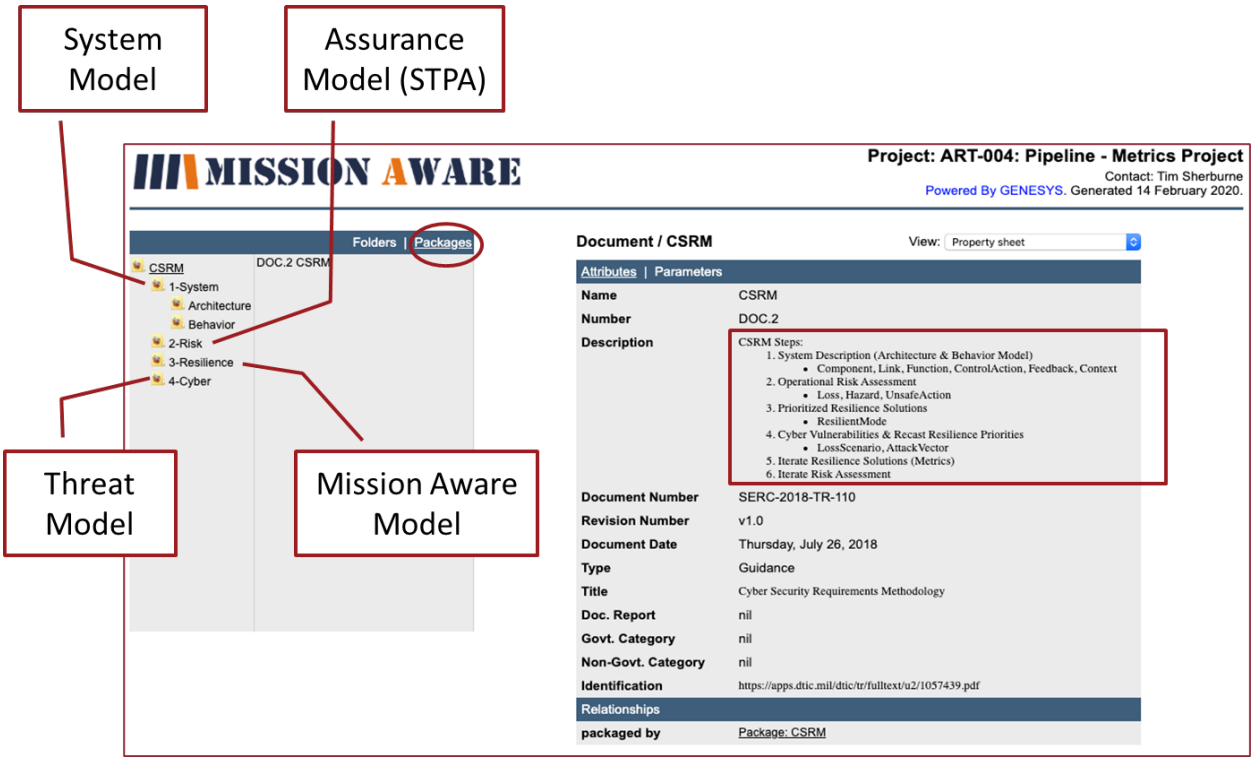

Figure 6. Main Navigation Page

The web-view model navigator, pictured in Figure 6, shows a package view to organize the model artifacts presented in this technical report. Expanding a package folder presents a hierarchy of related entity types. The System package defines the base System Model for the system under examination. Artifacts of the system description include the system context, the architecture of the system, and its functional behavior. The Risk package captures the Assurance Model (assurance cases), expressed in terms of losses, hazards, and unsafe actions. The Resilience package captures the system components and behaviors added to the system's base model that implements its resilience modes of operation, what we call the Mission Aware system. The Cyber package links loss scenarios to specific cyberattack vectors in an integrated Threat Model. These together form the MA Meta-Model. Further elaboration on the publicly accessible model is in the Oil & Gas Pipeline Model section of the report.

RESEARCH APPROACH

The project attempted to conduct a fully simulated development lifecycle for an engineering effort to "design-in" resilience in a complex system of systems. The effort started with a scenario development effort to describe a realistic cyberattack on a critical system and associated cost relationships to both attacker and defender. The "Black Monday" scenario (an attack on oil and gas pipeline infrastructure) was selected from this process based on realism and richness of attacker/defender trades. It represents an Advanced Persistent Threat (APT) scenario in a critical cyber-physical system that could be modeled. Figure 7 shows the flow of critical activities to derive assurance requirements and design patterns. Step 1 was the scenario development activity. Step 2 developed a threat CONOPS to express the APT complexity in engineering terms to aid in analyzing system versus threat interactions to form a cyber-physical systems (CPS) security requirements methodology with assurance case modeling. Steps 3 and 4 followed the standard SERC CSRM and MA Meta-Model process, creating an integrated system, threat, and resilience mode in an MBSE environment. Steps 5 and 6 were partially completed to assess the feasibility of using the meta-models for engineering tradespace analysis and requirements derivation.

In the rest of the document, assurance standards and modeling methods for resiliency will be discussed. The team reviewed the various assurance standards and the modeling methods for resiliency and building in assurance. The findings are reported, and the chosen methods are discussed. Synopses on system security engineering and loss-driven systems engineering are also reported. The use case and the oil and gas pipeline model is also discussed.

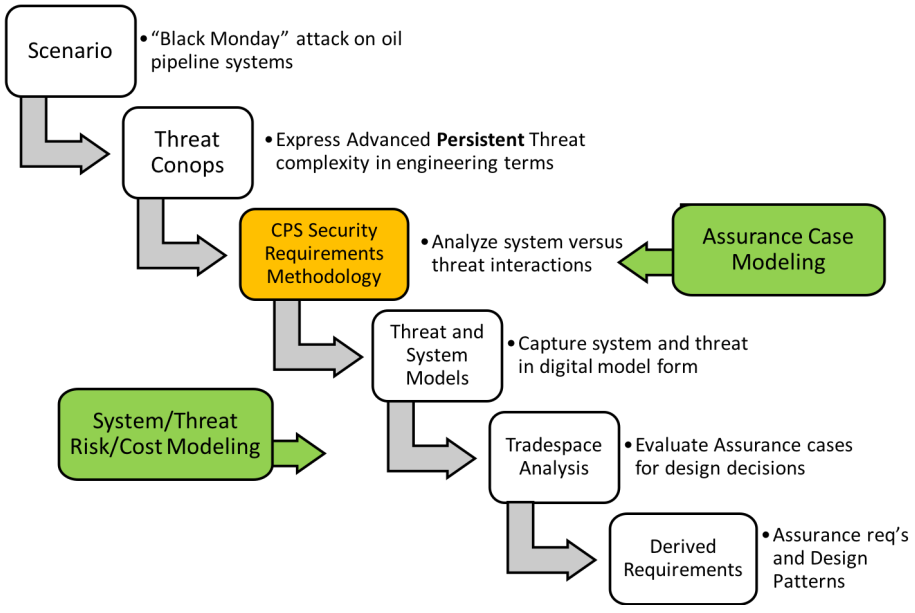

Figure 7. Project Activities for Security Assurance in Design

Modeling the system within the context of dependability, changeability, and resulting resilience is indispensable. The work done within the SERC on system qualities and ontologies, namely RT-46, 113, 137, 160, and 181, highlighted that when speaking to resilient systems, the complexity of requisite system attributes is evident. Resilience necessitates a design phase that ensures those systems are dependable and available in the presence of threats and other types of failures, resulting in them being reliable and robust [14] [15] [16] [17] [18] [19] [20].

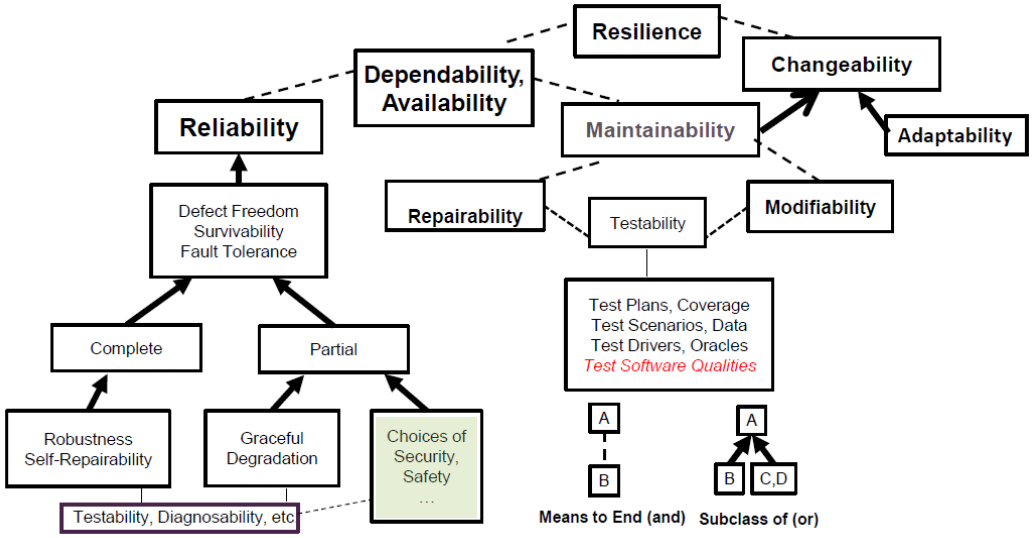

Figure 8. Dependability, Changeability, and Resilience

Implementation of safety and security is a set of design choices. The term maximum reasonable assurance is used to describe the output of that design. Designers cannot predict everything or assure everything, but they should achieve the maximum reasonable amount of assurance, given the cost, schedule, and other factors that drive programs. There are a number of different approaches to evaluate security and resilience. System security engineering is the starting point for security evaluation in systems engineering. Various assurance methods can then be used at different stages of the system lifecycle and these are discussed in the following sections.

SYSTEM SECURITY ENGINEERING

The National Institute for Standards and Technology (NIST) in November 2019 published Special Publication 800-160 Volume 2, Developing Cyber Resilient Systems: A Systems Security Engineering Approach. The publication is "a handbook for achieving the identified cyber resiliency outcomes based on a systems engineering perspective on system life cycle processes in conjunction with risk management processes." [21] It extends NIST Special Publication 800-160 Volume 1 Systems Security Engineering to the domain of cyber resilience, which in turn rests on ISO/IEC/IEEE 15288:2015, Systems and Software Engineering, Lifecycle processes. NIST SP180-600v1 emphasizes security engineering practices to protect against loss of assets [22]. NIST

SP180-600v2 addresses approaches to respond to cyberattacks but does not focus on losses. It defines cyber resiliency as "the ability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises on systems that use or are enabled by cyber resources." [21] Cyber-resilient systems have assurance measures built into the design as a foundational part of their architectures. An underlying assumption is that a well-resourced and persistent adversary will exploit design vulnerabilities and that these systems are designed to withstand fault, failure, and attack. New systems and planned modifications to or upgrades to existing systems should incorporate features that monitor faults, failures, and attacks and resilient modes that are enabled in repose to these. Cyber resiliency engineering practices are the set of MPTs that analyze and propose these system features and modes.

NIST 800-160v2 identifies five process steps in a cyber resiliency analysis:

1. Understand the context (threat, operational, architectural, and programmatic)
2. Establish a resiliency baseline (capabilities, gaps, and evaluation criteria)
3. Analyze the system (attack surfaces, vulnerabilities, threat opportunity, and opportunities for improving the system)
4. Define and analyze specific alternatives (system requirements, design, and trades)
5. Develop recommendations (analyze and assess alternatives)

In this research, we encode the cyber resiliency analysis into the CSRM process, which is discussed Appendix A.

Analysis should be conducted with a consistent metric, whether assessing threat context, mission, and operational context, architectural context, or programmatic constraints. NIST 800-160v2 does not recommend such metrics. However, this research recommends that loss and gain metrics can be consistently evaluated at all layers, or abstraction levels, of the systems engineering process. In the operational context, this results in a prioritization of loss scenarios and losses. In the threat context, this results in a scenario definition and possible quantification of threat gains. The contexts are dissimilar from established cyber resiliency metrics that focus on criticality or importance of system tasks or functions in the operational context, and exploitable vulnerabilities in the threat context. The dissimilarity is significant because metrics of loss and gain can be directly associated with the definition of risk and opportunity.

In contrast, criticality and vulnerability typically associate with the likelihood and consequence of individual risks. System loss metrics are an essential step in reasoning about system resilience at the operational and architectural level (section III. Use Case: Black Monday Scenario and the Oil and Gas Pipeline Model). As the goal of this research was to develop methods that better analyze cost and risk, this is an important starting place.

ASSURANCE METHODS TO INTEGRATE INTO SYSTEMS ENGINEERING

Structured assurance cases refocus requirements generation through claims and justification of those claims. Therefore, they can be a vital tool in overcoming the barriers between safety and security assessment methods and tools. Assurance cases were first presented in their current

form by Kelly, and capture the subjective judgment, which are the *what and why* of design choices [11]. In particular, this subjective judgment is structured through claims which must be supported by evidence. In addition to this basic structure, assurance cases make explicit the context, which includes (1) the system architecture; (2) the system environment; and (3) the system expected service.

Assurance cases currently host quality aspects of content, form, and structure with a semantic organization that should clearly state conclusions with explicit degrees of uncertainty. However, assurance cases are primarily informal and based on informal logic [23]. The purpose is not to replace methods and tools from reliability, dependability, and safety, but rather to reveal the relationship of such analyses with their higher-level claims within the design and acquisition process. The ineffective use of assurance cases stems precisely from assuming they are used as checklists to adhere to safety standards [11] [24]. Instead, assurance cases should be provided with concrete analyses and metrics to support the claims present in the assurance case.

The Structured Assurance Case Metamodel (SACM) is a standard specified by the Object Management Group (OMG). SACM provides a richer set of features than existing system assurance languages/approaches. SACM provides a foundation for model-based system assurance. Assurance cases are typically represented either textually - using natural languages, or graphically - using structured graphical notations such as the Goal Structuring Notation (GSN) or Claims-Arguments-Evidence (CAE). Graphical notations have gained popularity due to their abilities to express clear and well-structured arguments. Several tools exist which implement GSN and CAE. To improve standardization and interoperability, the Object Management Group (OMG) specified and issued the Structured Assurance Case Metamodel (SACM) [8].

Assurance, as defined, is grounds for justified confidence, gained before depending on a system, that a claim about dependability, safety, or security has been, or will be, achieved. A claim is a true-false statement about one of these properties of a system. Assurance is related to the "requirements of a property of a system." System assurance, as defined by NATO, is the justified confidence that a system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle [25]. System "functions as intended" and "free of exploitable vulnerabilities" represent the highest-level properties of the system. The cybersecurity community has focused too firmly on exploitable vulnerabilities. It needs a much more rigorous approach to gain confidence that the system functions as intended in the presence of external threats. This confidence is achieved by system assurance activities, which include a planned, systematic set of multi-disciplinary activities to achieve the acceptable measures of system assurance and manage the risk of exploitable vulnerabilities. One can argue that "functions as intended," for systems of any complexity, requires a modeling method that relates system function to the requirements properties of a system that define its dependability, safety, and security. These properties can be constraints on the system function or can be additional system functions that support assurance activities [26].

Model-based safety and security have a long history of starting with reliability and dependability models, and there are several studies for assurance cases for safety and security models. Most importantly, the team compiled examples and scoped the methods needed for inclusion to address cyber assurance in relation to systems engineering. Numerous safety assessment methods have been modified for use in uncovering security vulnerabilities and understanding security risk [27].

The team evaluated assurance methods for implementation into the modeling of the scenario by assessing the

1. closeness to reality,
2. abstraction,
3. interoperability,
4. tool support, and
5. verification and validation.

In addition, the different assurance methods, currently, loosely map to different timelines during the system lifecycle. The proximity to the stage of the lifecycle addressed, reality, abstraction/level of fidelity, interoperability, tool support, and verification and validation were most closely met by STAMP – STPA and STPA-Sec. The unification of the STPA technique for safety analysis and STPA-Sec for security analysis has proved to guide in-depth security analysis to the most vulnerable and critical components of a system. A synopsis of the examination can be found in the literature review, an ancillary document available upon request.

For future research, it could be useful to integrate more assurance cases since there is room for improvement in linking several methodologies to create a better-understood system within the context and organization.

SYSTEM THEORETIC PROCESS ASSESSMENT (STPA)

To address the shortcomings of linear failure models, Leveson developed the System-Theoretic Accident Model and Process (STAMP), and System Theoretic Process Analysis (STPA). STAMP has been applied to accident analysis and prevention and general dependability and security design in CPS using a security-specific form of STPA called STPA-Sec [28]. In the STAMP framework, understanding system disruptions requires analysts to determine why the control structure was ineffective. STAMP replaces the concept of an event that results from a control failure with the concept of a constraint that enforces appropriate control. STAMP analyzes and imposes an equivalent structure of CPS information and control feedback. This feedback extends hierarchically from a central control structure to include larger feedback loops created by more extensive system dependencies. The prospective interactions between dependent systems are changed from assumed trust to evidence of trust. Evidence of trust allows the system to have adaptive feedback loops that either maintain or fail to maintain system characteristics of dependability and security [29].

The design of CPS resilience begins with the identifications of system hazards and vulnerabilities. The identification is a human process that is naturally limited by the knowledge of the human teams involved, so it should be an iterative process that evolves with the System design and use. The first step in any design for safety programs should be identifying the system hazards, where accidents must be defined for the particular system being developed [26]. An accident need not involve loss of life, but it does result in some loss that is unacceptable to the customers or users. For practical reasons, a small set of high-level hazards should be identified first. Even complex systems usually have fewer than a dozen high-level hazards. Starting with too broad a list at the beginning, usually caused by including refinements and causes of the high-level hazards in the list, leads to disorganized and incomplete hazard identification and analysis process.

A system is a recursive concept, meaning a system at one level may be viewed as a subsystem of a more extensive system. Hazards, or unsafe behaviors, at the system level can be translated into hazardous behaviors at the component or subsystem level. However, the reverse (bottom-up) process is not possible. Hazards can also be related to the interaction between components such as the interaction between attempts by air traffic control to prevent collisions and pilots' activities from maintaining safe control over the aircraft, due to the potential vulnerabilities (i.e., threats to a system's intended safe function) stemming from an array of complex interactions and sequences of events, STAMP views accidents – by analogy to the CPS security case, threats – as a control problem. Vulnerabilities may consequently be prevented by enforcing certain constraints on system component behaviors and their interactions. In STAMP, a process model controls the actions to help expose what is deemed an unsafe control actions: control commands required for safety are not given, unsafe control commands are given, commands are given too early or too late, or the control action stops too soon or is applied for too long [29] [30] [31].

This view implies a modeling approach that captures the functional state space of a CPS and reveals whether that state space has been compromised or preserved in the face of threats and applied protection patterns. The focus on functional modeling espoused in this report is consequently synergistic with the goals of STAMP. Further, the frameworks and methods discussed may serve as a direct complement to existing model-based systems engineering processes and tools and, in turn, themselves be executable within a toolset that enables systems engineers to produce, navigate, and understand the complexity and scope of the problem.

STPA-Sec takes the mission objectives, and operational tasks developed in the war-room setting and extend them down to dependability and security objectives. The first step is the identification of unacceptable losses in the mission/operational context. This list is then used to derive a set of system hazards and associated system constraints. Take, for example, a simple digital engine control loop in an aircraft. Unacceptable losses would include loss of the aircraft, loss of human life or injury, unacceptable delays in travel, and loss of trust in air travel. Three (of many) specific hazards in an engine controller would include uncontrolled changes in engine thrust, incorrect engagement of engine thrust reversers, and incorrectly reported engine failures. An example system constraint would include a requirement such as "the system shall prevent engagement of thrust reversers while the aircraft is in flight."

This high-level descriptive model can then be used to create a functional model of the control structure, leading to a set of expected control actions and potentially hazardous control actions. The thrust reverser example if relatively simple. Two control actions likely define the function: engage and disengage reversers. Defining this would be accompanied by several constraints defining functions such as "check for weight on wheels before engaging…" All of these would be typical system functions in a system functional model. The STPA process recommends evaluating the causes and effects of not providing the expected control action, providing it in a way that causes hazards, providing it too soon or too late, and providing it out of sequence. Detailed examples of STPA and STPA-Sec can be further explored in the open literature.

STPA-Sec extends the analysis to identifying hazardous control actions and related security constraints, such as "thrust reversers shall not engage without direct physical indications of aircraft weight on wheels" or "weight on wheels indicators shall employ diverse redundancy." This process leads to scenarios that relate hazardous CPS control actions to security-related scenarios (and dependability-related scenarios as a set of control constraints). These can be further explored in the context of threat attacks and associated system loss of control (expressed as errors) using tree or graph models [26] [31].

LOSS-DRIVEN SYSTEMS ENGINEERING

Security, safety, and resilience (and associated dependability attributes of systems) can be explored in an integrated process focused on concepts of loss. The resilience of a system 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 [2]. In this context, system assurance is a loss-driven methodology for identifying and evaluating resilience alternatives and balancing the effectiveness and affordability of system design alternatives. It considers four modeling goals:

1) a model of the system and its mission, operational tasks, behaviors, and structure
2) modeling the concept of maximum reasonable assurance - the decision process that considers system performance safety, security, dependability, and associated characteristics and determines the appropriate responses to malicious and non-malicious disruptions to the system that could result in losses
3) models that capture engineering rigor – the engineering methods and processes that support the specification, architectural definition, design, analysis, and verification & validation of the system, and
4) creation of a system resilience model – a model that communicates the system, the threats to the system (disruptions and resultant losses), the assurance decisions (requirements and constraints), and the countermeasures (design decisions and resilience modes) added to the system model.

The research focused on capturing all four modeling goals in a consistent environment using Model-Based Systems Engineering (MBSE) methods, processes, and tools. A primary outcome of this research is the development and maturation of a meta-model capturing primary concepts of

the system (operations, function, structure, requirements), assurance (loss, loss effect, and loss scenario), and resilience design (functions to avoid, withstand, recover, and adapt) into MBSE tool constructs.

The CSRM process and the STPA methodology underlying it begins with identifying high-level losses that system owners and users desire to avoid. STPA postulates possible ways for accidents to occur with or from the system that result in loss, then analyzes and facilitates the development of requirements and design alternatives that makes those accidents and losses less likely to occur. Loss scenarios are core analysis tools. The process is different from traditional assurance activities that use a claim/argument/evidence format. The traditional justifies claims that the system operates as intended (meets its performance expectations), though systems that are operating "as intended" can still result in losses (particularly in the presence of cyber advanced persistent threats). An analysis of loss focuses on effects, where traditional assurance processes are more focused on the cause. Thus, loss-driven systems engineering offers an approach for higher-level reasoning about a system and the consequence of its behavior (meets acceptable loss expectations), both intended and unintended. The combination of capability-driven and loss driven systems engineering provides a more robust analysis of full system operation in all contexts. Reed and McEvilley propose a Systems Engineering method of analysis for loss that is “specialty independent” and has application to all assurance activities, which is shown in Figure 9 [3].

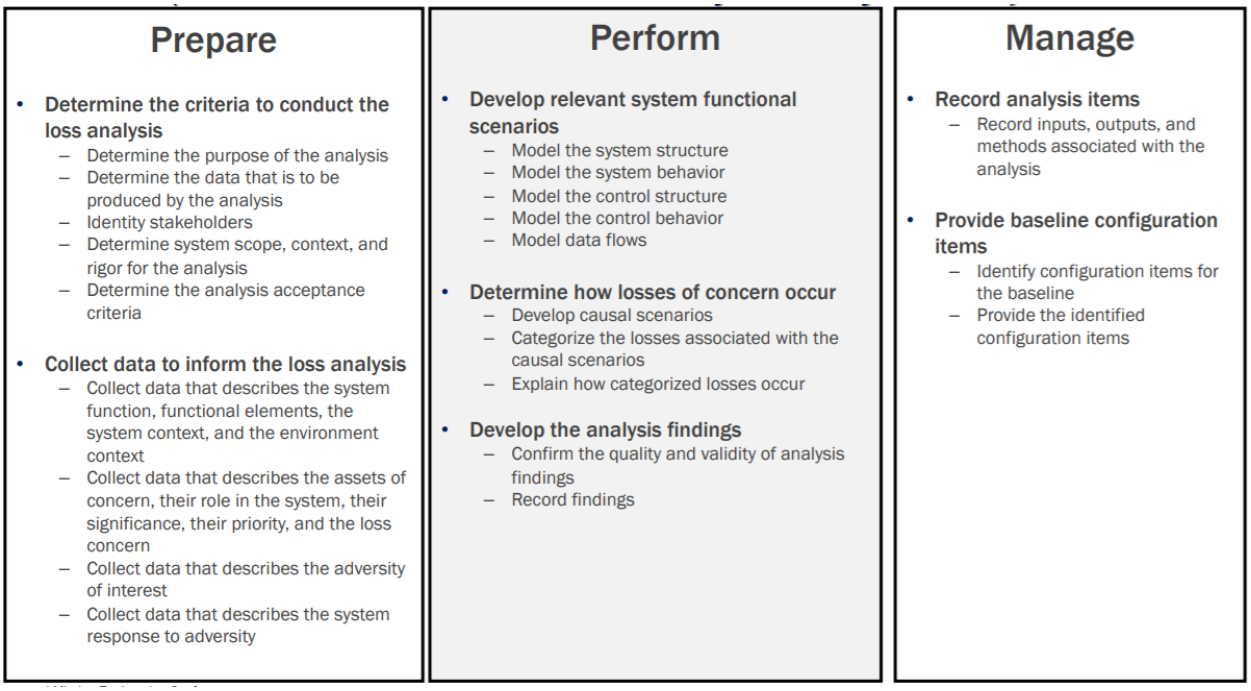

Figure 9. Loss driven systems engineering analysis [3].

ASSURANCE CASES

STAMP/STPA belongs to a class of accident causation models (ACM). The use of STAMP/STPA should result in a more robust identification of system hazards in the early stages of system definition and requirements setting. STPA provides a natural form when evaluating resilience modes as these link constraints on system and controller operations (usually captured as requirements) with the architecture's resilience features. Figure 10 shows this linkage [32].

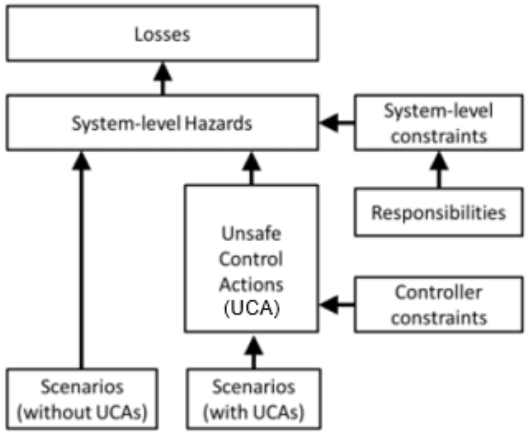

Figure 10. Traceability between losses, hazards, and unsafe control actions in STPA [32].

An assurance case in this relationship would take the form of a claim that constraint “x” would eliminate or reduce the probability that an unsafe control action would result in hazard and then loss. There is a natural linkage between methods in collecting evidence through modeling and testing that this claim is valid. It is not clear based on our research that assurance cases in a formal argument level offer anything over STAMP/STPA at higher levels of abstraction. However, at lower levels of design, an assurance case language that formally specifies the design can be useful. The Resolute assurance case language developed on the DARPA HACMS project takes a higher-level architecture model. It establishes a set of lower-level design arguments that can be formally modeled and tested.

Assurance case design for cybersecurity cases is addressed in the DARPA High-Assurance Cyber Military Systems (HACMS) and Cyber Assured Systems Engineering (CASE) programs but still has limited acceptance across the community, specifically since assurance is a latecomer to MBSE. The HACMS program goal is to create technology for the construction of high-assurance CPS, where high assurance is defined to mean functionally correct where safety and security properties are fulfilled. Fundamentally different from the software community's approach, HACMS adopted a clean-slate, formal methods approach to producing semi-automated code from executable, formal specifications. CASE's goal is to develop the basic design, analysis, and verification tools to allow systems engineering to design-in cyber resiliency and manage tradeoffs [33] [34] [35]. However, further research can be done to link AADL and Resolute. There is a higher-level system behavior in the MA Metamodel that can be adjoined with the lower-level design models to specify the system's entirety formally.

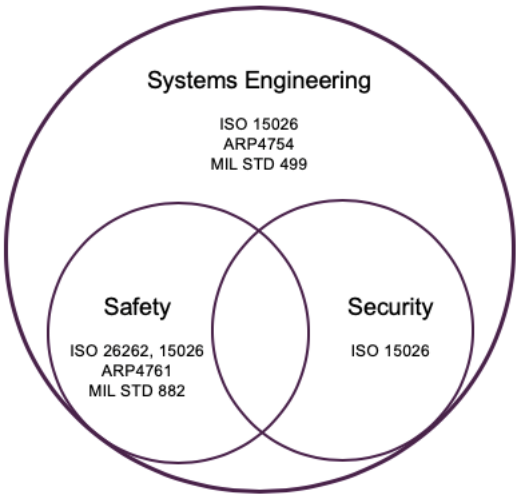

Figure 11. Relationship between Systems Engineering, Safety, and Security

Software and systems assurance, and closely related fields, share concepts but have different vocabularies and perspectives. The ISO 15026 seeks to provide a unifying set of underlying concepts with an unambiguous use of terminology. Assurance is “grounds for justified confidence, gained before depending on a system,” and is related to “requirements of a property of a system.” To delve further, “often a property is specified as a behavior.” The behaviors could

be restrictions on allowed system states, system states that must be reached, or constraints on flows or interactions. In the ISO 15026, an assurance case is a reasoned, auditable artifact supporting the top-level claim. The structure and content have claims, arguments, evidence, justifications, and assumptions. There are safety, security, and dependability cases involving safety, security, dependability, or RAM (reliability, availability, and maintainability) [10].

CYBER MISSION ASSURANCE ENGINEERING

Cyber Missions Assurance Engineering provides a risk-based methodology for identifying and evaluating alternatives for reducing cyber risks concerning the effectiveness, efficiency, and affordability of alternatives. Alternatives can be evaluated at the enterprise, mission, and system tiers, and for programs which may acquire multiple related systems, services, or infrastructures. As illustrated in Figure 12, the Cyber MAE methodology consists of five component processes: (1) establishing mission priorities, (2) identifying mission dependencies on cyber resources, (3) performing a mission (or business) impact analysis, (4) performing a threat susceptibility analysis, and (5) analyzing alternative cyber risk remediation alternatives for effectiveness, efficiency, and affordability.

Mitigations can be drawn from multiple disciplines, including assurance, security, resiliency, anti-tamper (AT), and supply chain risk management (SCRM). Practices and techniques in these areas counter or otherwise address the threats to the target system, program, or mission. Mitigations can vary in effectiveness, maturity, and cost. Thus, the risk remediation analysis considers these, as well as constraints imposed by organizational culture, policy, legal and contractual limitations, and commitments to technologies or standards.

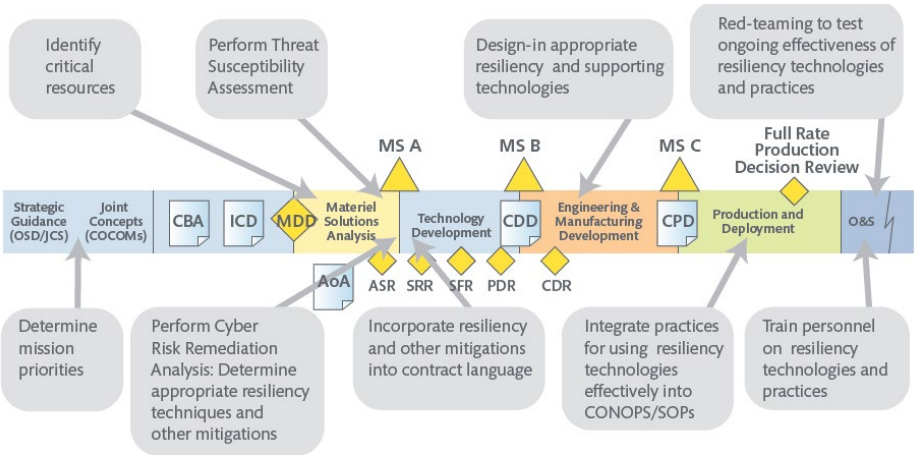

Figure 12. MITRE MAE Methodology

Based in this research, formal models of loss and gain provide additional insight into the cyber mission assurance activities because they can be assessed consistently across a larger portion of the pre-Milestone A lifecycle. Using Model-Based Engineering (MBE), the relationships between mission priorities, CONOPS, threat susceptibility, risk analysis, and resilience features and be formally captured to produce system requirements or constraints. MBE is defined as “an

approach to engineering that uses models as an integral part of the technical baseline that includes the requirements, analysis, design, implementation, and verification of capability, system, and/or product through the acquisition lifecycle." [36] At a higher level, Model-Based Systems Engineering (MBSE) is described as "the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later lifecycle phases" [37]. A system model represents structure, behavior, requirements, and parametrics (equations). In formal MBSE, these aspects are coupled. MBSE's strong emergence elucidates the opportunity for assurance in all lifecycle stages of the systems design process, and movement from document-centric operations to objective and mission-based products that are embedded and compatible with the model in the MBSE environment.

Model-Based Assurance, in turn, aims to serve system verification and validation requirements. MBA requirements are typically determined via the development of formal assurance cases. Assurance cases specify primary verification and validation requirements for the system design process and provide an explicit means for justifying and assessing confidence in critical properties. Verification and validation of mission-critical systems through test and evaluation have historically been the gold standard for assurance. However, they are significantly expensive and increasingly fraught with difficulty as systems – CPS, in particular – become more complex, expansive, and expansive interdependent on other systems to realize their intended capabilities [38] [8]. The model-based approach supports red-team activities at all stages of the system definition and development, not just at the end.

In the digital engineering (DE) effort, the use of models to support the analysis of assurance has seemed promising. Still, it faces challenges to establish systems engineering foundations to produce evidence to support assurance judgments. It is especially true for CPS, which is often employed in SoS operational configurations. They are increasingly connected and progressively complex in the context of their higher-order dynamics with other systems in the environment and face increasingly diverse and sophisticated threats. MBA and MBSA are sets of SE activities that use a model, or group of models, as a basis of understanding to produce evidence that a given system will perform as intended in various potential environments, operational conditions, arrangements with other systems, etc. [39] [40]

FORMAL ASSURANCE MODELING APPROACHES

Although not fully evaluated in this research, the CSRM process and MA Metamodel will be useful for tracing higher level mission, operational, and system architecture level analyses into more detailed design analyses linked to formal assurance case languages. The MA Metamodeling approach in this research explores functional methods and models for security analysis and design. This approach may link nicely with the High Assurance Cyber Military Systems (HACMS) project conducted by the Defense Advanced Research Projects Agency (DARPA), which explore formal methods [34] [33]. However, the systems engineering research community has not established clear boundaries for application of functional analysis and evaluation versus formal analysis and evaluation in assurance design. Formal methods refer to the application of a fairly

broad variety of theoretical and formal models to problems in software and hardware specification and verification and are being studied for extension into more complex problems like the large-scale embedded systems like the military employs. The point where the transition from more highly abstracted system functional and executable models to detailed design models of assurance characteristics becomes necessary is not well researched [41].

The effective threat coverage of the MA Metamodel will be limited by the ability of the human teams to effectively envision and model loss scenarios. It will also be limited by the high level of abstraction developed in the functional descriptions. The MA Metamodel will generally produce the requirements and high-level functional and architecture designs. A lower level architectural design model is needed for automated (mathematical) analysis. This model needs to reflect the details of the system software interfaces and execution flow. The systems model captured in a SysML language form can be linked into a domain specific model of the computing system and software to capture the requisite detail. The DARPA HACMS program selected Architecture Analysis and Design Language (AADL) [42] and associated modeling tools to define the system architecture and requirements models.

Formal methods create on provably correct implementation of system hardware and software components (system assets) modeled in a hierarchical architecture. Component level specification of the system is needed so this level of assurance analysis will be limited in the earlier concept definition phases of the system. Assurance builds from layer to layer – if the components fully satisfy their behavior and are free from vulnerabilities, and the system execution fully correctly implements the components, and the architectural model of the system correctly specifies the structure of the system and its interfaces – then the system can be mathematically proven to operate correctly in the presence of cyber threats. Formal methods differ from functional methods in the way the system is analyzed. In a functional approach, the functional behaviors are evaluated using tools like the MA Metamodel. The actual built system is analyzed or hypothesized with respect to its activities and resilience modes are designed into the system. In a formal approach the built system design is analyzed as flows of function or information through component hierarchies. Tools that evaluate correctness of implementation mathematically are used to ensure the built system matches the assurance specifications.

The DARPA HACMS project created Resolute [43] – a formal assurance case evaluator - for conducting formal security analysis of the CPS components and architecture layers. AADL is a domain specific language that supports modeling of typical computer system hardware and software in a hierarchical format, along with the as-designed information flows and activity threads. Resolute is a semantic assurance case analysis tool that can decompose high level assurance case arguments down to the component level to evaluate full system assurance. It implements a form of GSN linked to the AADL architectural language to provide a full assurance design environment.

The shortfall of one method versus the other is the assumptions made in the design process. For example, proving a hardware component is free from vulnerabilities in a formal process often assumes that resilience modes have already been applied to lower level components. A purely functional analysis only addresses the activities in the information and processing flow and does not ensure that components are built correctly. Both methods suffer from overall ability to analyze

as the system scale increases. Based on this discussion, one can see that there is a natural linkage between higher level accident causation models like STPA and lower level assurance case languages like Resolute. The value of this research is to move the formal analysis of assurance to a point much earlier in the system lifecycle, when the architectural decision space is more open. Additional research is recommended to formally link the CSRM and MA Metamodel artifacts to the AADL tool suite developed by DARPA, creating a full lifecycle analysis capability.

RESULTS AND DISCUSSIONS

The top-level arguments of the research are the following.

- We can develop a standard, model-based approach to reason about assurance across the domains of safety, security, dependability, and RAM, and across the full system lifecycle.
- We can implement the assurance concepts and evidence in MBSE tools.
- We can associate these with the resilience of a system at any level in an environment with natural and malicious threats.
- We can then, in turn, determine metrics at various levels of the system to support decisions about maximum reasonable assurance and sufficient resilience.

The CSRM and MA Meta-model, as defined in this research, provide a standardized approach to link modeling, assurance cases, system constraints (requirements), and what we term "resilience modes" (additional system functions) in an MBSE toolset. The MA Meta-model provides a standard set of design patterns to formalize this approach.

USE CASE: BLACK MONDAY SCENARIO AND THE OIL AND GAS PIPELINE MODEL

To better address the primary goals of the research task, a candidate system was developed for a use case. The use case had to be robust with evident cost and benefit motivations within an international cybersecurity incident. The assault also had to be that of an advanced persistent threat with gradual gains. From there, the team researched the domain, compiled evidence of vulnerabilities, and then produced a model within GENESYS to further exploit security assurance pitfalls.

BLACK MONDAY SCENARIO

The team worked with graduate and undergraduate students in a course focused on scenario development and wargaming at Georgia Institute of Technology, centered on international cybersecurity incident scenarios. The students were provided direction to develop scenarios based on cyber-physical systems, where the realized value to the attacker could be quantified. The students crafted original security scenarios that concentrate on disruption or subversion of cyber-physical systems. The students then evaluated cost/benefit motivations within the scenarios and then matured them through gaming exercises. One scenario was selected from the four student scenarios. Titled the "Black Monday," the scenario and represents a unique example of an APT in a critical infrastructure system exploited for monetary gain, in this case, pipeline oil pumping stations and associated pipeline oil delivery operations and market activities.

The team selected this scenario as it 1) represents a plausible APT in a critical infrastructure managed through both human operator and cyber-physical control systems, 2) the relative scope of the threat team's effort can be estimated, and the monetary gains from the attack can be modeled, and 3) it represents an exploitable gap in existing security practices in large systems tied to multiple organizations in the supply chain. Also, the architecture of pipeline oil pumping activities has a scope that can be modeled in present MBSE tools. A summary of the scenarios follows:

"Black Monday" is a posited a cyber-attack executed on Saudi Aramco Riyadh & Yanbu, Baiji (Iraq), and SPC refineries. Fancy Bear, (which is a Russian hacker group) gains remote access to the companies' refinery controls and reports false flow rates, pressure, and temperature of trunk lines. The Russian refineries report similar spills as time goes on, and report that a malicious code was "found" within their system. The Russian refineries then claim to solve the irritation plaguing the countries and offer world-class cybersecurity services and packages to all three countries. However, backdoor measures are installed for future purposes and are targeted at manipulating critical pipeline pumping stations in the pipeline networks bringing oil from source to refineries. At some point later, a separate attack is executed on pipeline pumping stations across three major pipelines. These attacks result in degraded flow, causing a yield of oil decreasing by 6.2m barrels/day. The students estimated that the disruption could last for 30-60 days. These three major pipelines carry approximately 10% of the world's crude oil. An overall 10% decrease in global oil availability was "gamed" to result in a 50% price increase per barrel in global markets, which for 30 days was estimated at $31 billion total market impact. The students further posited that a state-driven attack would potentially reap substantial gains in the oil futures market, potentially generating enough revenues to make up for recent significant losses in the Russian economy driven by low oil and gas prices. The scenario is particularly interesting because 1) it represents a realistic threat from an APT in a critical infrastructure system, and 2) the losses and benefits that accrue from executing the attack can be quantified.

The modern oil and gas cyber-physical systems (OGCPS) incorporate information and communication technologies to improve broad area control, maintain situational awareness, and control physical processes remotely. Therefore, the above scenario is not far-fetched. Primarily provided Russia's standing with oil and software packaging as exports. The industrial control systems (ICS) are key within the systems operations and how the cyber components affect the physical components and system mission. A multilevel attack incorporating numerous segments with the persistent threat on both the SCADA and ICS is most feasible.

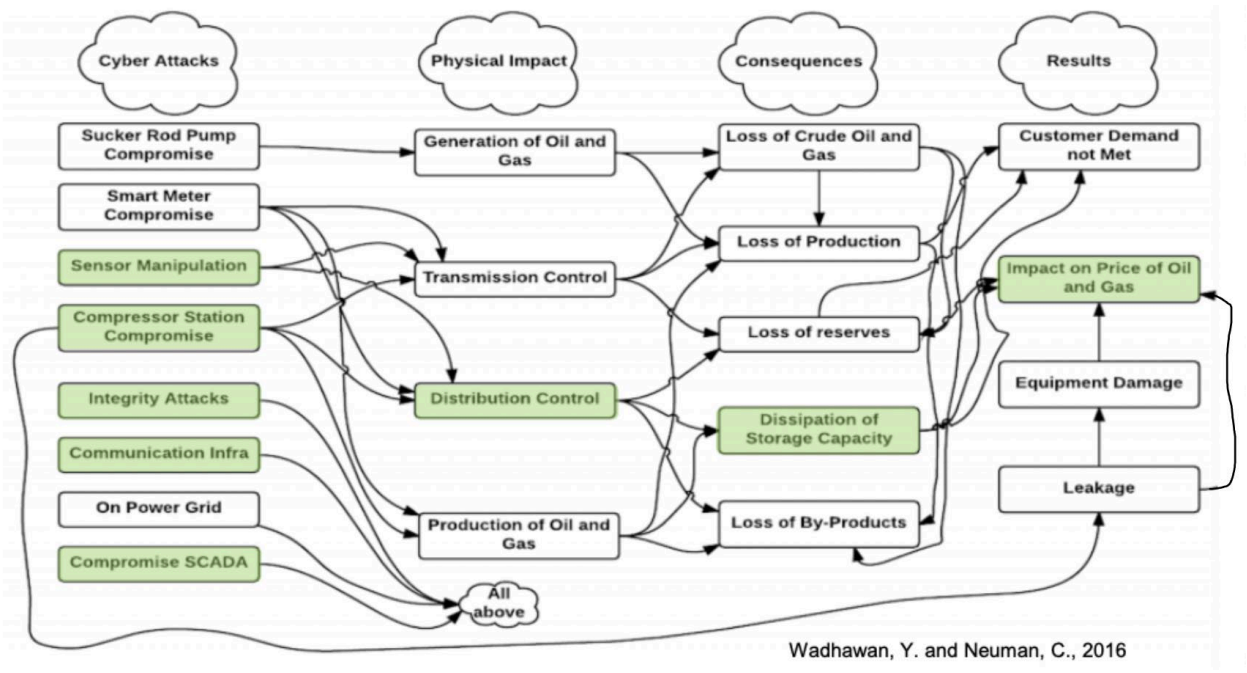

Figure 13. OGCPS Attack Diagram [44]

The coordinated attack would be on multiple oil pumping stations on the following systems: pump control, pipeline monitoring, distributed control system, leak detection, control centers, and maintenance operations. When and where the attack is initiated and propagated is critical for an advanced persistent threat to capitalize on a distributed stakeholder set where there is a lack of system-of-systems view [44].

THREAT CONOPS

To capture the complexity of this scenario concerning real oil pipeline systems, the team elected to conduct and document its research on the systems and attack vectors using a Concept of Operations (CONOPS) document format. The CONOPS from this project is intended to inform mission descriptions that are critical to the lifecycle process and various stakeholders for the oil and gas cyber-physical system (OGCPS), which consequently also provides crucial information to adversaries for the capitalization of interdependencies within the operations. From the company operation's perspective, the document is an amalgamation of several publicly released artifacts from various organizations, companies, and academic papers. The attacker's perspective is from literature reviews, experience, and curated and created models that can infiltrate the infrastructure.

CONOPS documents usually describe a proposed system's characteristics from the viewpoint of an individual who will use that system. It is a user-oriented document that describes system characteristics for a proposed system from the user's viewpoint and communicates both quantitative and qualitative characteristics. The CONOPS should also inform the sets of capabilities that can achieve the desired specific objectives and end states. This CONOPS,

however, belabors the threat operations of the specified system. Therefore, the CONOPS is split into the following highlighted sections.

1. Normal Operations
2. Maintenance and Support Operations
3. Opportunities and Constraints on Change
4. Threat Operations

An APT is a broad term describing a cyber attack where hackers covertly gain access to a system and remain inside it, undetected, for a significant period to achieve a specific goal. While usually an APT is backed by a nation-state or state-sponsored group, it has recently referred to non-state sponsored groups conducting large-scale targeted intrusions. Often, there are several fixed assaults to cloak the adversary and achieve the objective. An APT in a critical system incorporates the social engineering (research, data and credential harvesting), physical engineering, vulnerabilities (SCADA zero-day), push rogue logic, and executed outcomes (or a lack of predictive models). Figure 14 provides a general vision of the attack process from a published summary of an actual attack on critical infrastructure control stations. The attack scenarios must include operations to gain access to the system (through phishing – left side of the diagram) and exploit access to the system (right side). The scenarios also include staging targets (top) in which the threat learns about and plans the attack and the actual targets of the attack (bottom). This project simulated the staging activities in the development of the CONOPS but only explored the actual execution through our model.

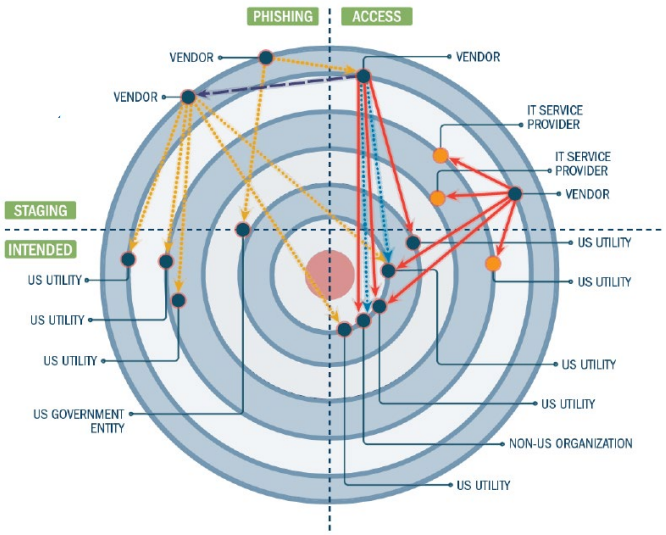

Figure 14. Advanced Persistent Threat Target(s) [45]

Understanding the adversary and its operations to achieve specific goals is essential to effective cybersecurity and assurance. There are various paths and patterns of attack to percolate through a system. Within the CONOPS, the vulnerabilities are highlighted and paired with known patterns of attacks employed by adversaries. The research utilized a list of known attack patterns is from the Common Attack Pattern Enumeration and Classification (CAPEC) database and demonstrated some vulnerabilities in the model, which were realized through the Common Vulnerabilities and

Exposures (CVE) database, both from MITRE. The Common Weakness Enumeration (CWE) was also scoped to best understand the specific software or hardware weaknesses within the OGCPS. This help build the threat patterns (loss scenarios and attack patterns) in the model.

The Threat CONOPS guided the modeling and discussion for the use case, enabling identification of causal factors that produce a loss effect to generate the loss scenario. The constituted elements and relationships that result in loss-informed analysis can determine response action and its efficacy as well as inform risk management activities. The effort also has the foundations to start characterizing a system's mission evolution: to examine built mission threads and prioritize to ascertain impeded mission impacts from APTs and help work from threat to security controls.

OIL & GAS PIPELINE MODEL

Oil and gas pipeline companies share the concerns of policymakers and others regarding the potential implications of a security violation on industry assets. There are ongoing activities to protect critical infrastructure, provide reliable energy for society, and preserve public safety and the environment. Adversaries to this industry activity include nation-states, criminal organizations, and unaffiliated bad actors seeking to steal intellectual property, compromise industrial control systems, and other nefarious goals. The industry has seen the evolution of cybercriminals and the advancement of the techniques, tactics, and procedures they use, moving from manual operations to more sophisticated and widespread machine-to-machine automated attacks with the use of augmented intelligence. There are multiple other attack vectors, including insider threats and attacks via supply chain tampering or disruption. This pipeline model shows how the metamodel captures the system's mission-oriented functions, the potential losses to the stakeholders, and the implications of security violations.

This demonstration system is decomposed and organized according to the mission aware methodology using the Vitech GENESYS MBSE modeling tool, which was extended with our metamodel. The particular tool is not necessary to use the metamodel, but we use the tool and its associated diagrams to visualize the different model views as defined by the metamodel. The tool provides a mechanism to export a system model to a web-based team view. In this section, we present a summary of significant model artifacts, but we offer the complete model as opensource, which does not require to use the tool. The full pipeline model is available online [https://github.com/coordinated-systems-lab/pipeline-cps].

The web-view model navigator (Figure 15) shows a package view to organize the model artifacts presented in the following sections. Expanding a package folder presents a hierarchy of related entity types. Based on the metamodel, the system model's general overview takes the form of a system description, which contains architecture and associated behavior, the operational risk to the system, resilience design patterns based on the operational risk, and potential threats to the system. In the following sections, we will navigate the model at each level to show how the metamodel assists with adding structure to the modeling activity and how it relates to these different but essential views. We will not focus on an exhaustive presentation of the model itself

but rather show the relationship between the model and the metamodel and how it could be used to analyze and extend the system's safety, security, and resilience considerations.

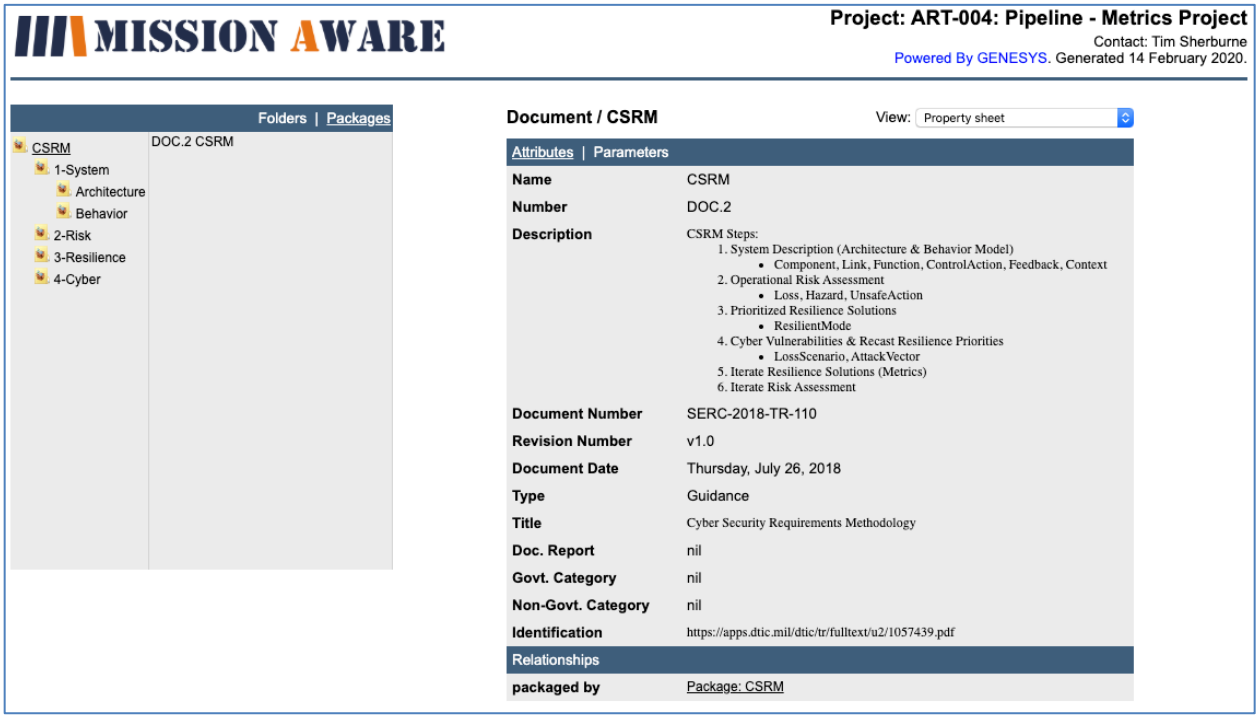

Figure 15: Team View Main Navigation Page

SYSTEM DESCRIPTION

The system description defines the base model for the system under examination. Artifacts of the system description include the system context, the architecture of the system, and its functional behavior.

The system context physical block diagram (Figure 16) defines the boundaries and external interfaces for the oil and gas pipeline being evaluated. Each node on the diagram is an instance of a *component* while each connecting line is an instance of a *link* associated by the *connects to* relation. The system context diagram enables a common understanding among stakeholders of the scope of the system model. In our demonstration system the pipeline endpoints (drilling rigs, refineries, etc.), wide-area communications network (cellular, satellite, etc.), environment (weather conditions, geography, etc.), and security operations center, for sharing of industry-related events, are all external to our demonstration system model. Additionally, to enable simulation of cyber-attacks, an advanced persistent threat interface is included. Within the system context is the pipeline itself and a peer mission aware system (sentinel) which is concurrently evaluated and is responsible for resilient mode reconfiguration based on detected illogical system behavior which may indicate safety and/or security vulnerabilities.

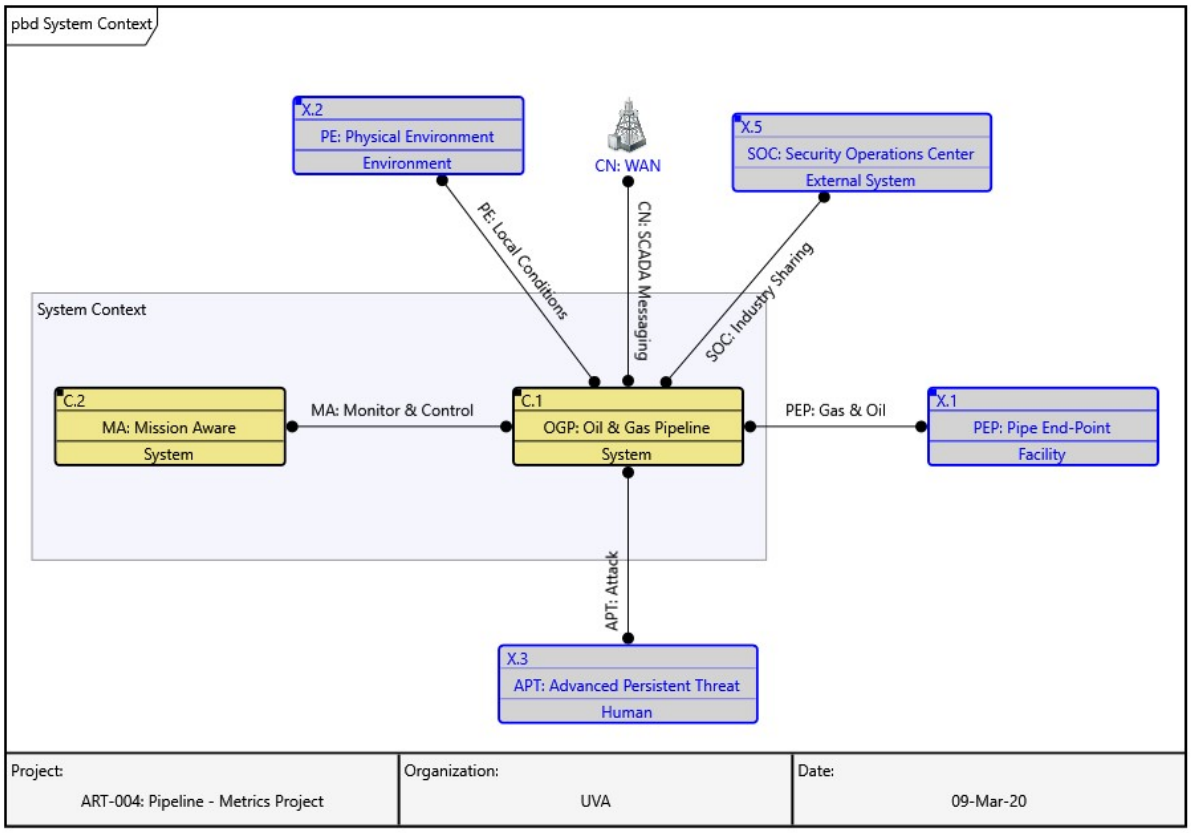

Figure 16. System Context Physical Block Diagram

The system context block definition diagram (Figure 17) shows three levels of the pipeline system decomposition. Each node on the diagram is an instance of a *component* while each connecting line is a *built from* association between components, showing the cardinality of each sub-component. In our demonstration system, the oil and gas pipeline contain a single human operator, one or two (if redundant) system managers and n pipeline routes. In turn, a pipeline route contains one or two (if redundant) route managers, one or two (if redundant) route LANs, and n pipeline segments.

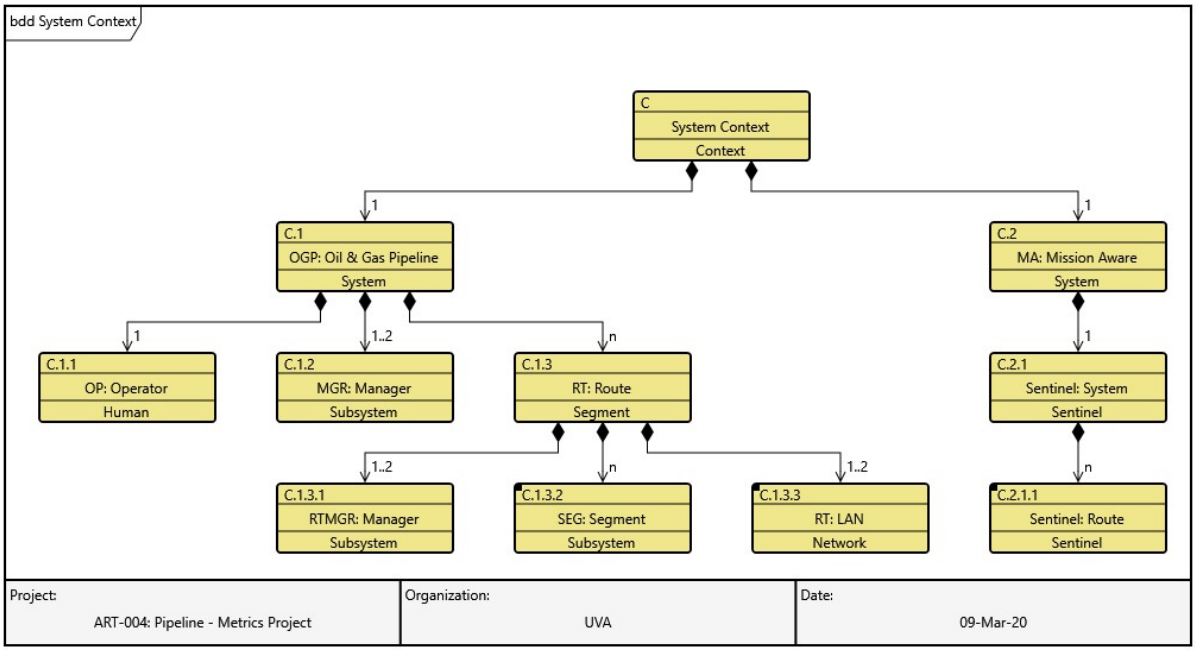

Figure 17. System Context Block Definition Diagram

Finally, a physical block diagram (Figure 18) defines the architecture of a pipeline segment. Each segment includes the physical pipe, pump, and valve that deliver oil and gas from one endpoint to another. A set of sensors (pressure, temperature, flow rate) provide feedback to the segment controller on the operational state of the segment. A segment LAN connects segment components for SCADA messaging and also provides connectivity to the higher-level route LAN. As part of the regular maintenance process for a pipeline segment (known as “pigging” within the industry), a smart PIG is periodically used to clean and inspect the pipe.

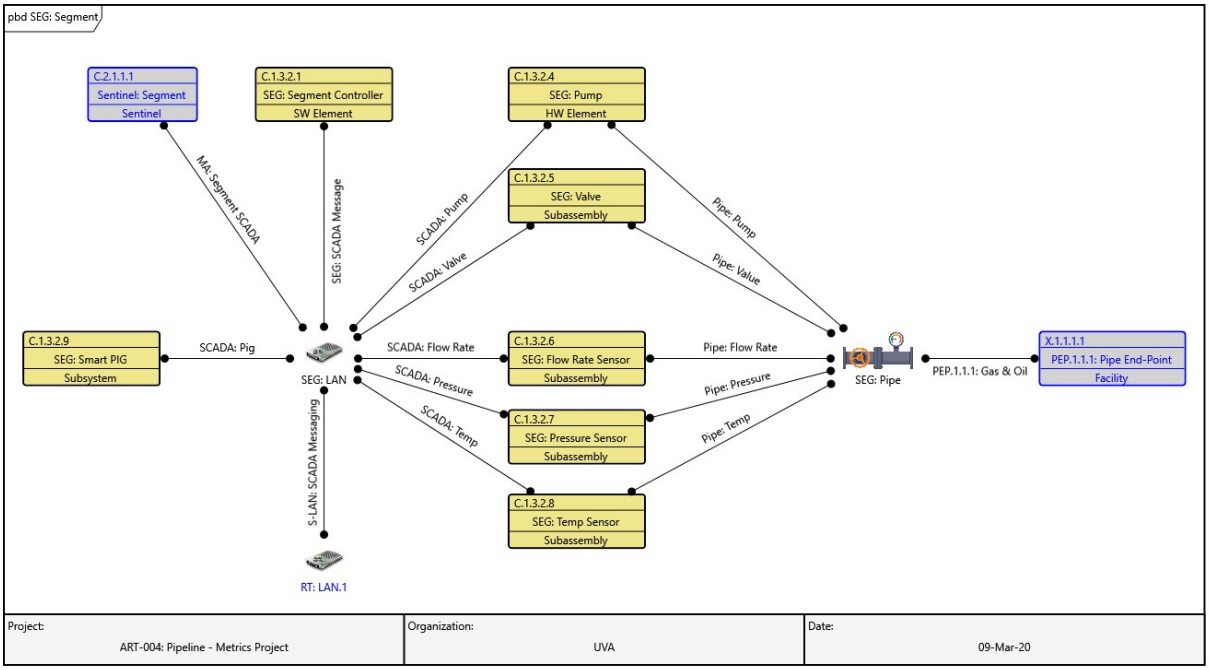

Figure 18. Segment Physical Block Diagram

Turning to system behavior, an enhanced functional flow block diagram (EFFBD) (Figure 19) defines the top-level behavior for the oil and gas pipeline in the form of a feedback control structure. The diagram shows the top-to-bottom hierarchy of the control structure. The outer-level *and* block shows that each of the lanes of behavior (operator, system manager, route manager, segment controller) execute in parallel. The behavior control flow logic is captured as instances of the metamodel *call structure item*. Each of the yellow rectangles represent an instance of a metamodel *function* showing the function number, name and *performed by* associated *component*. The green rounded rectangles represent instances of a metamodel *control action* or *feedback* item. The lines between functions and control actions or feedback items are represented as *outputs* or *triggers* associations. The *replicate* blocks for the Route Manager and Segment Controller indicate that there are n concurrent instances of these behavior blocks and are captured as call structure items that decompose their respective branches of the behavior model.

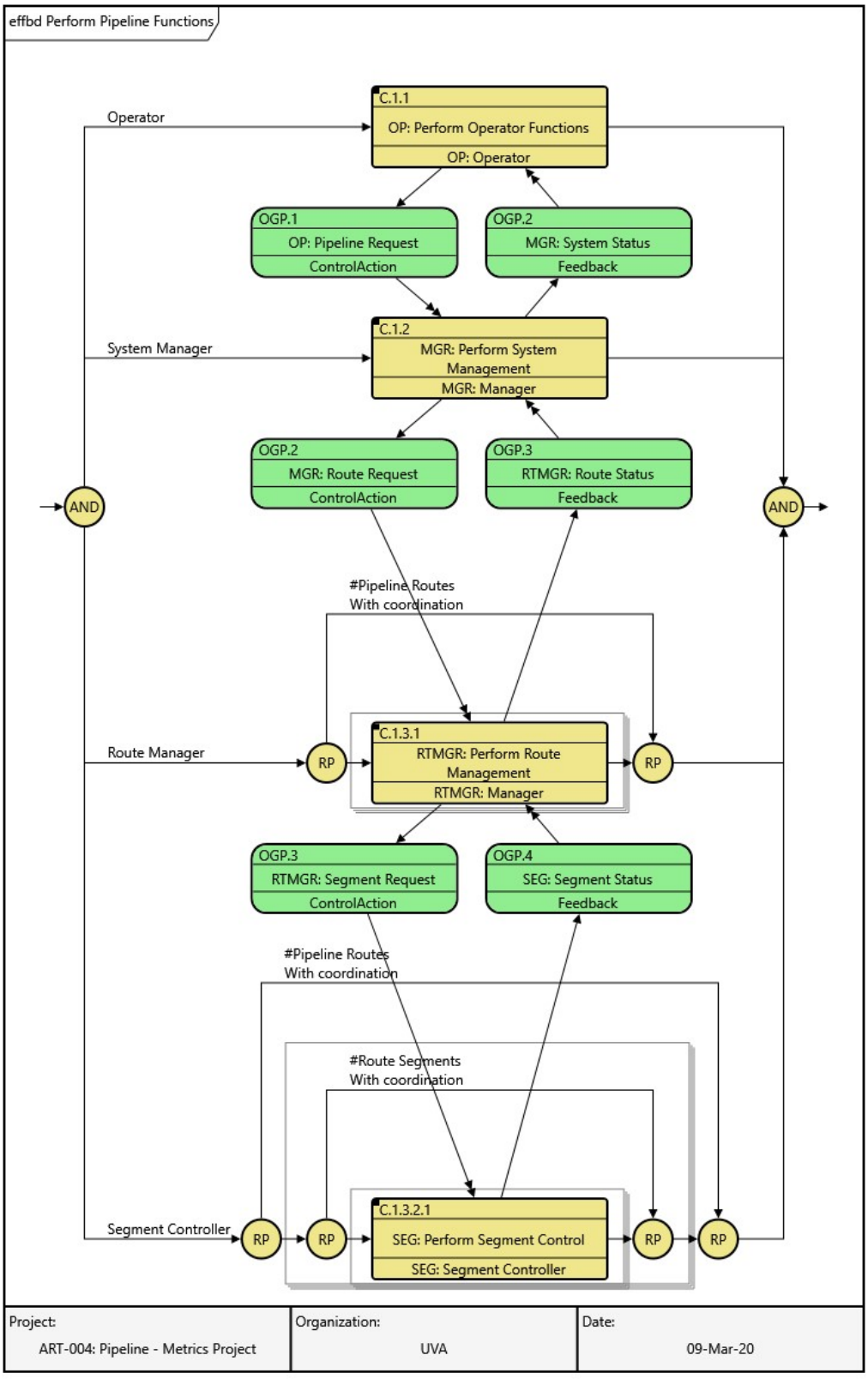

Figure 19. Top-level Pipeline Behavior Diagram

The *perform segment control function* is further decomposed by a second level EFFBD (Figure 20) with lanes for operations, control, status, and transfer. Each of these lanes operates in parallel in a continuous loop. The handle route manager request receives control actions from the route manager and maintains the requested state within the segment status: *context*. Based on segment state context (requested state and senor status), the control pipe pump and value function initiates control actions to the segment pump and value. The *collect pipeline sensor status* function monitors and maintains the sensor status within the segment state: *context* and then forwards that state via segment status: feedback to the route manager. Finally, the *transfer gas and oil* function provide the physical movement of oil and gas through the pipe segment.

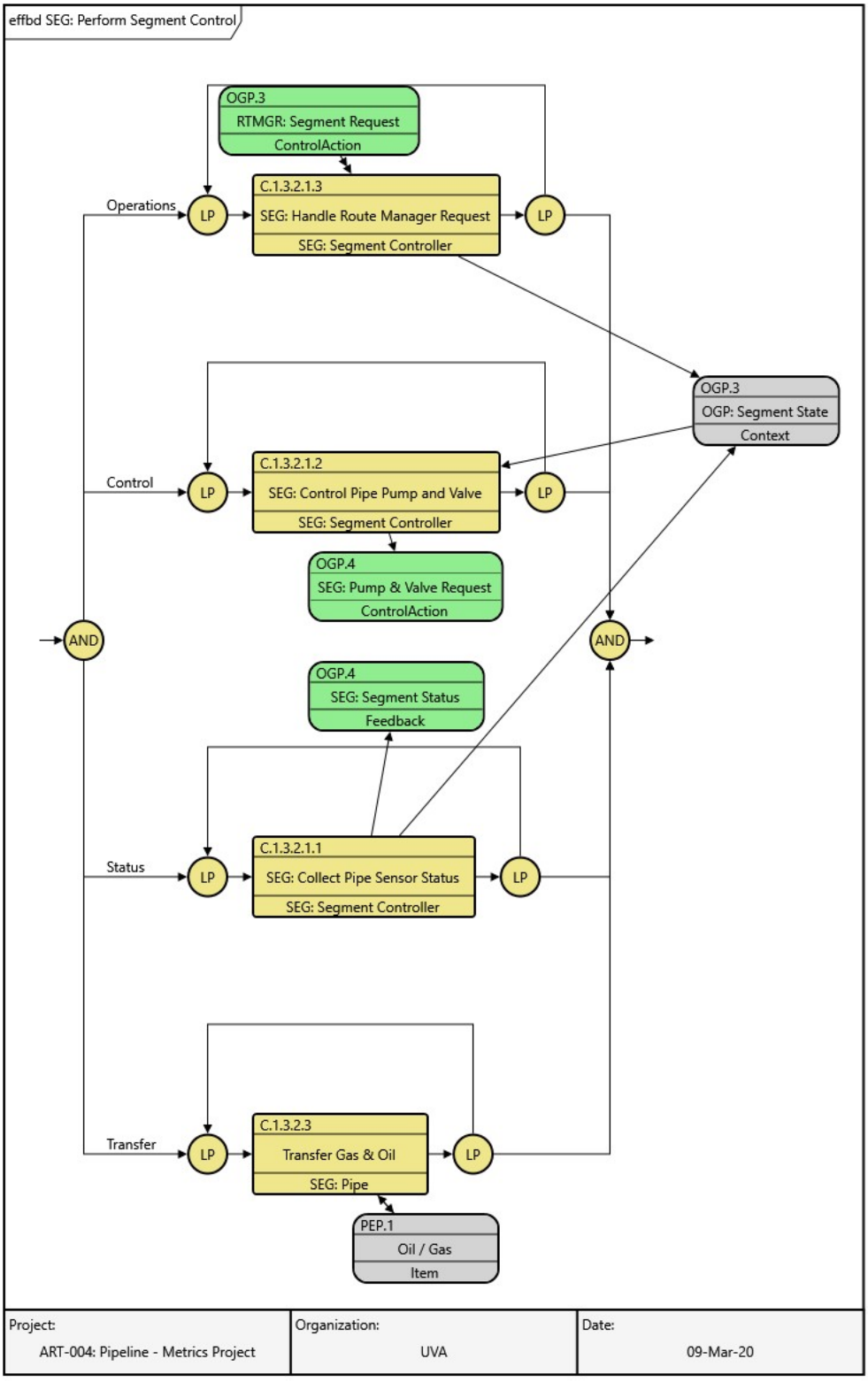

Figure 20. Segment Behavior Diagram

After the system is described to an appropriate level of detail, attention is focused on understanding operational risks for the system from the perspective of end users. Through STPA the metamodel provides a top-down process to aid in the identification of the model artifacts for the operational risk assessment including *losses*, *hazards*, and *unsafe actions*. The system behavior model provides an inventory of *control actions* that are methodically considered to identify potential *unsafe actions*. As defined by STPA, there are four ways a control action can be unsafe:

- Not providing the control action leads to a hazard.
- Providing the control action leads to a hazard.
- Providing a potentially safe control action but too early, too late, or in the wrong order.
- The control action lasts too long or is stopped too soon (for continuous control actions).

One example of such risk is *decrease route flow* (Figure 21) where flow rate for a pipeline route is decreased before achieving optimal flow as defined by the associated process model context route state. This unsafe action is *a variation of* the set route flow rate control action which can *lead to* equipment operated outside normal specification hazard which in turn can *lead to* either equipment damage or sub-optimal capacity losses. If the users require a tradespace analysis, it is required that system operators prioritize losses. Losses are the main criterion for operational risk and by prioritizing losses it is possible to compare and contrast what safety, security, or resilience considerations should be added to the system first.

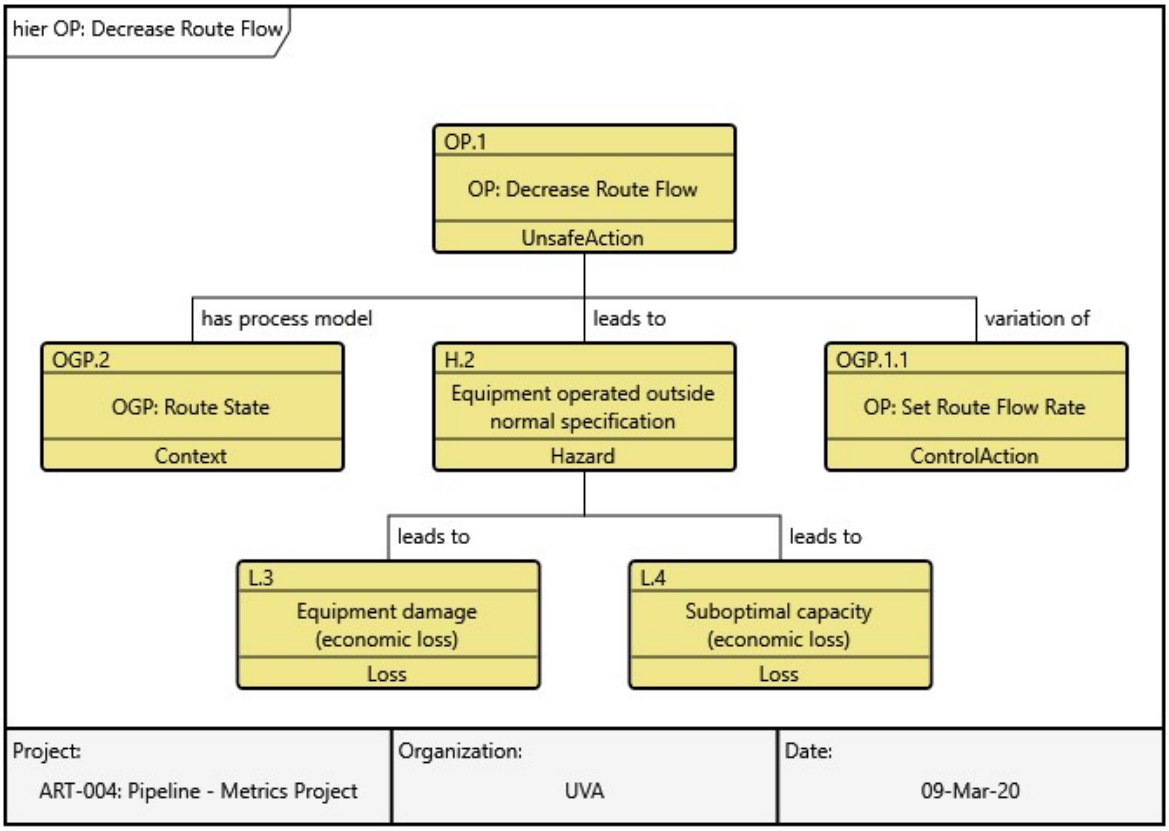

Figure 21. Pipeline Unsafe Control Action

RESILIENCE SOLUTIONS

Following the operational risk assessment, system *resilience solutions* are proposed by system design experts. These solutions are focused on segments of the system that are within a feedback control path for related unsafe actions that lead to the highest priority system losses. Design patterns for resilience solutions include diverse redundancy (which limits the effectiveness of insider or supply chain attacks), hardened design, perimeter defense, etc. An example of a resilience solution is diverse redundant sensors (Figure 22) where the segment controller and sensors are contained by the solution. The degree of *contained by* associations are an indication of implementation complexity and provide an additional aid in tradespace analysis. The solution *recovers* the collect pipe sensor status function and *is managed by* (enable or disable) the *perform segment contro*l function. The control function could be performed by an operator or, if desired, automated by a sentinel.

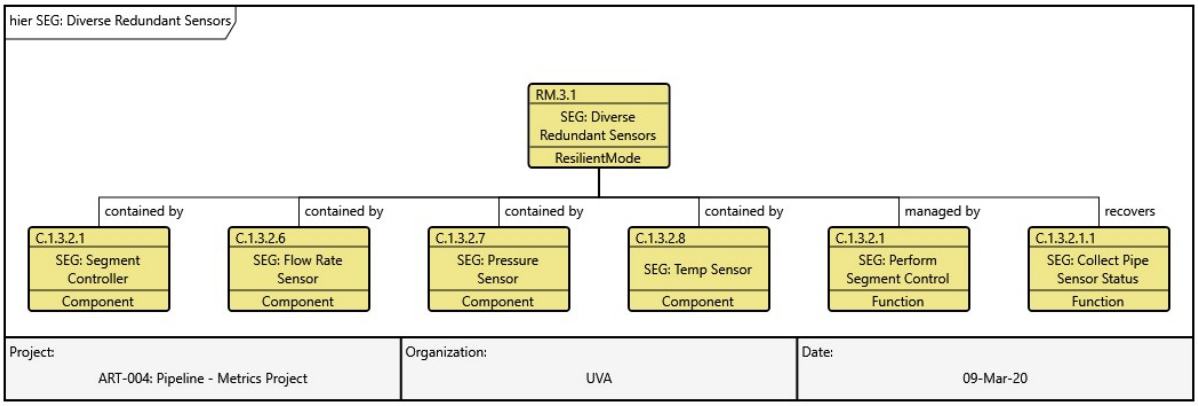

Figure 22. Pipeline Resilience Solution

VULNERABILITY ASSESSMENT

Guided by safety and security experts a system vulnerability assessment is performed next. Identification of *loss scenarios* is the primary metamodel artifact that captures this assessment. STPA provides a structured approach for identifying loss scenarios by analysis of the system feedback control structure while security experts will also consult databases of *attack vectors* considering how a loss scenario could be *precipitated by* these attack vectors. The loss scenario is linked to the operational risk assessment using the *leads to:* unsafe action association and to a resilience solution using the *is remediated by*: resilient mode association. To enable a sentinel monitor, a *detected by monitoring* association to a system link, resource or function is defined. An example of a loss scenario is false sensor reports (Figure 23) where the loss scenario *is detected by monitoring* SCADA message: link, *is precipitated by* modification during manufacture: attack vector, *is remediated by* diverse redundant sensors: resilient mode, and can *lead to* decrease route flow: unsafe action. The vulnerability assessment is also responsible for determining appropriate values for the likelihood and severity of attack vectors as well as time budgets for detecting, isolating and restoring the system.

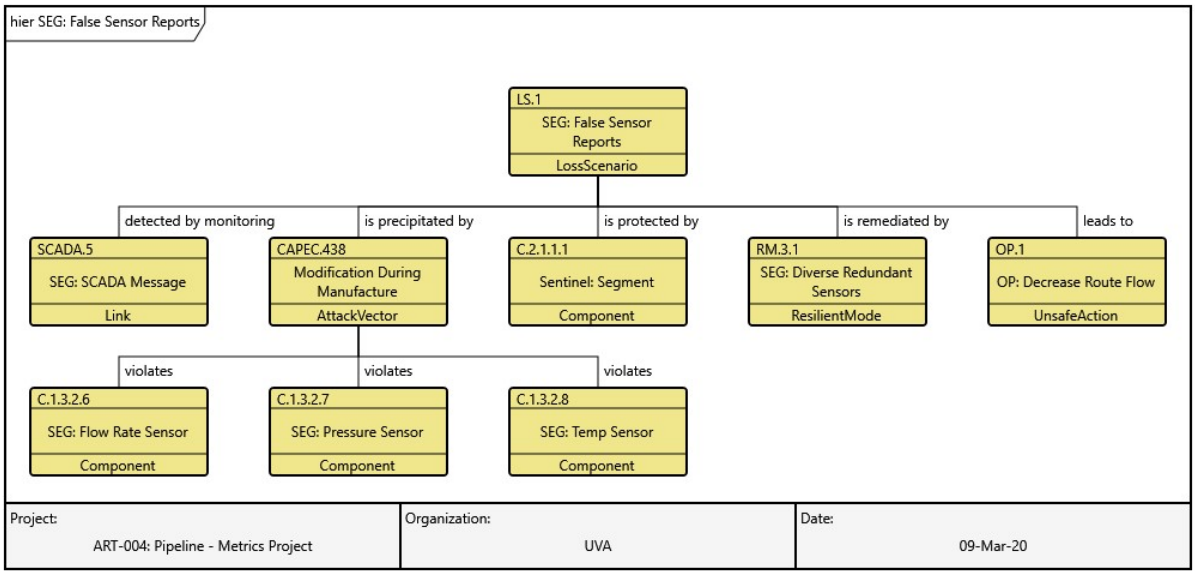

Figure 23. Pipeline Loss Scenario

ITERATIVE TRADESPACE ANALYSIS

Systems engineering is an inherently iterative process and balancing the perspectives of operational risk and system vulnerability also require iteration to achieve an optimal system solution within programmatic budget and time constraints. Example resilience metrics (Figure 24) provide a framework for evaluating the effectiveness of resilience solutions in response to safety and security violations while achieving operational priorities. The metamodel relates the expert and operator perspectives, which are required for priority ranking of system losses, likelihood and severity determination for attack vectors in order to evaluate the effectiveness and complexity of resilient modes. Other important evaluation metrics for system resilient modes are an understanding of the operational impact and the time budget for system recovery. Recovery time includes detection time, isolation time and restore time including any operator decision time. System simulation can evaluate the recovery ratio for critical system functions under various system loads and simulated attack patterns. Tradespace analysis, based on resilience metrics, enables specification of a system which responds to safety and security violations, while achieving operational priorities, within programmatic cost and time constrains.

For example, determining an appropriate resilience solution for a critical subsystem will likely have multiple approaches (redundancy versus hardening, etc.) with security experts preferring one approach while system operators possibly preferring another due to usability considerations. The meta-model provides a mechanism for all stakeholders to understand the trade-offs and a place to document agreements and the process used to reach consensus. This documentation is invaluable to future system enhancements, evolution and maintenance which likely involves different team members. As another example, a system architecture supporting a diverse

redundant subsystem must assure that the system recovery time budgets are met. A polling-based detection mechanism may prove to be insufficient and may instead require an event-based notification solution to achieve the required detection time under various system loads.

Object	Metric	Values	Notes
Loss	missionImpact	High / Med / Low	Blue Team
Loss Scenario	attackLikelihood	High / Med / Low	Red Team
	detection Time	seconds	Time budget to detect loss. Impact tradeoff for Sentinel interfaces: • polling-based (system / link loading) • event-based, etc.
	isolateTime	seconds	Time budget to isolate loss via system /component tests.
Resilient Mode	complexity	High / Med / Low	Number of model "contained by" associations. Indication of cost.
	effectiveness	High / Med / Low	Impact on remediating High "likelihood" attacks associated with High "mission impact".
	operationalImpact	High / Med / Low	Degree of operator training need. Degree of mission interruption.
	restoreTime	seconds	Time budget to restore system function via resilient mode. Impact tradeoff for Resilient Modes: • Active/Active • Active/Standby (Hot / Warm / Cold)
	operatorDecisionTime	seconds	Time budget for operator decision time to enable resilient mode. 0 implies automated resilient mode.
Function -> RecoveredBy	recoveryRatio [per Loss Scenario] *Calculated:* Measured / Expected	< 1: Acceptable > 1: Not Acceptable	Recovery time includes: • Detection • Isolation • Restoration Including: • Technical: System Components • Operational: System-of-System Interactions • Operator: Expected Decision Times

Figure 24. Resilience Evaluation Metrics

The DoD strategic vision for safety and security engineering seeks to achieve stronger synergy in the engineering approaches and methods of system and security safety [1] [3]. It is based on unacceptable loss effects with safety-relevance and/or security-relevance. Safety and security synergistic systems should address all forms of loss scope. For safety, the assurance would be freedom from those conditions that can cause death, injury, occupational illness; damage to or loss of equipment; or damage to environment. Assurance in security is freedom from those conditions that are aforementioned and include damage or loss of data or information; or damage to or loss of capability, function, or process. Key activities for synergistic system security analysis includes planning, analysis, and design. In addition, understanding the system need, structure, adversity, and behavior enables design for assurance. Integrating assurance standards and modeling methods should also link a metrics framework together with threat motivation and system loss trades. The research looked at loss-driven systems engineering and the current resiliency frameworks for cyber mission assurance.

The oil and gas case study determined resilience metrics that provide a framework for evaluating the effectiveness of resilience solutions in response to safety and security violations while achieving operational priorities. The metamodel of the use case relates the expert and operator perspectives, which are required for priority ranking of system losses, likelihood and severity determination for attack vectors in order to evaluate the effectiveness and complexity of resilient modes. Other important evaluation metrics for system resilient modes are an understanding of the operational impact and the time budget for system recovery. Recovery time includes detection time, isolation time and restore time including any operator decision time. System

simulation can evaluate the recovery ratio for critical system functions under various system loads and simulated attack patterns. Tradespace analysis, based on resilience metrics, enables specification of a system which responds to safety and security violations, while achieving operational priorities, within programmatic cost and time constraints. The use case, modeling, and Meta-Model showed the feasibility of achieving this for a given system. The research shows results for a resilience evaluation metrics framework that links together threat motivation with system loss trades, but further progress is needed to formalize.

COST AND RISK ESTIMATION

While the relationships between mission-level resilience and the design of the subsystems remain poorly understood, ART-004 begins to establish practices that could contribute to better cost modeling and risk assessment for the discernment of security in design. In the development of Systems-of-Systems today, one must account for the various costs of interwoven layers of security from the mission level down to the network and mission system architecture levels. The costs of mission cyber resilience are often simplified for evaluation of risk to the weapon system. However, adding security features increases design and recurring production costs in hardware and software systems. Also, most military systems have interrelated security and safety concerns. Metrics are needed to assess the quality of different requirements and design solutions based on safety and security risks in the presence of a determined cyberattack. System methods are needed that trade requirements and design decisions based on the evaluation of hazard/risk (loss of equipment, property, damage to the environment, death, injury, and occupational illness), cost, and an understanding of the threat, which requires timely, accurate, credible and relevant threat information.

Currently, many research methodologies are still siloed so that the subsystem relationships and interdependencies do not propagate through the system to ensure the mission is upheld in the presence of an adversary. Part of the inspiration for this research comes from work presented by Wortman et al. on modeling adversarial risk in cybersecurity scenarios [53]. Ideally resilience metrics should incorporate both the design/cost decisions on the system itself and the design/cost decisions an attacker might make. The approach taken in this project was to simulate that process using a system modeling team and an adversary modeling team. Per the CSRM process, system models are between system design and adversary design as we want to find as many potential loss/gain decisions (and associated risk/reward decisions) as possible. This research defines an approach to determine a “security risk” metric. As security risks are based on a series of external and internal vulnerabilities and exploits, security risk probability cannot be determined by analysis of singular vulnerabilities. An analysis needs to aggregate cost of loss/benefit of attack. A descriptive model is quite useful for this. This research takes the traditional Risk = Probability * Consequence (Impact) equation and extends it to a method that considers probability of attack, probability of successful attack, and impact of successful attack. Simulating these three factors in the MA Metamodel, one would determine probability of attack via loss scenarios, probability of successful attack via resilience mode implementation, and impact of attack as a ranking of unacceptable losses. This research did not try and calculate these

metrics for the oil and gas scenario as a full design process was not completed. A full model-based design effort could calculate these relationships from the process. The component costs of the system as it is designed would have to be known and calculated. Referring back to Figure 25, a loss scenario cost and resilient mode implementation cost would be added to the table.

Another method that does attempt to estimate security by loss of each stakeholder to the mission success is applying the Cyber Security Econometrics System (CSES) to Mission Assurance Analysis Protocol (MAAP). CSES has a cascade of linear model approach to measuring attributes that support security countermeasures. The framework has three steps: generation of stakes matrix, dependency matrix, and threat matrix. The stakes matrix acknowledges the system's stakeholders, while the dependency matrix considers the architecture and components of the system. The preliminary refinement, due to the connections not being developed, qualitatively suggested the seven guidelines of MAAP aligned with the CSES framework [46]. However, no mathematical analyses were conducted to estimate mitigation costs.

According to the MITRE Cyber Resiliency Engineering Framework, published in 2011, three types of cost can be used to define and characterize cost metrics, representing organizational and operational concerns in the areas of initial costs (I), support costs (S), and consequential costs and benefits (C). The initial costs are the dollar or level-of-effort costs to establish the resiliency techniques, including the development, acquisition, and integration costs. The support costs are the dollar, or level-of-effort, costs necessary to maintain and facilitate the effective use of the approach. Support elements can include the CONOPS development, policy development, testing, and risk assessment. The consequential costs and benefits are the result of using the approach and can apply to all of the various stakeholders. This can include change to the CONOPS, and amount of resources applied to other mission support components [47]. This approach maps well with the approach taken by Wortman et al.

Within ART-004, the initial costs can be preliminarily estimated with insight into the development and support costs captured into the development of the CONOPS. While no actual advanced persistent threats were initiated on the oil and gas pipeline, the research and design process to develop the CONOPS and system architecture models aided in understanding the adversary effort required to exploit the system. An adversary such as Fancy Bear, would use a similar process to determine their series of exploits. The CONOPS development puts the systems engineering team holistically into the mind of the attacker. This overcomes a limitation of traditional attack tree analysis which drives the focus of the analysis down to individual attack vectors and component level vulnerabilities. While the information is spread across several different outlets, it was easily compiled to expose vulnerabilities and create a thorough threat Concept of Operations allows exploitation of the system.

While APTs used to cost several hundred thousand dollars, the ease and ability have dwindled the price significantly. However, when targeting critical infrastructure, understanding the standard operations, maintenance and support operations, the environment in which it lives, and the opportunities and constraints on the system expound the vulnerabilities for an APT to attack sufficiently and cause the desired outcome.

The research in ART-004 increases the credibility of the MA MBSE Meta-Model as an effective path toward security assurance in early stage design. In addition, it has the foundation for more formalized methodologies addressing operational risk assessment, resilience evaluation metrics and solutions, and vulnerabilities assessments in early concept definition phases of an acquisition program. It is also a flexible approach, allowing for new insights to be integrated as the system shifts phases within the lifecycle. The STPA methodology paired with the CSRM creates assurance arguments and evidence at the architectural level that the MA-MBSE Meta-Model can then capture and maintain through a model-based design process.

This research combined operational scenario development, Conops development, and use of the MA-MBSE Meta-Model to demonstrate an overall approach to evaluate cost/technical risk and opportunity decisions for security assurance in design. It enabled the team to capture specialty perspectives in an integrated architecture model using MBSE tools and established the credibility of the model as an effective path toward security assurance in early stage design. The general approach developed in the work serves as a basis for a repeatable, yet flexible, approach. The framework and foundations developed in the research are ready for transition. A particular transition focus is toward mission engineering and early stage system definition in the government MBSE modeling settings, but the techniques can and should be applied consistently across all program lifecycle phases. Modeling assurance cases and resulting resilience modes of the system is a key aspect of system architecting and the MBSE MA Meta-Model provides a standard architectural representation for loss scenarios, assurance requirements, and resilience features of the architecture. Assurance cases are intended to be developed and maintained for the full lifecycle and the MBSE MA Meta-Model provides a standard approach to capture all aspects of the assurance process. Further information on the MA-MBSE Meta-Model is in Appendix A.

FUTURE WORK AND RECOMMENDATIONS

The efforts within ART-004 accentuated gaps, opportunities, and barriers within MBSE and MBSA for security assurance in design. While preeminent progress was made, future work will still need to mature respective aspects of the research as well as continue to expand to integrate with mission engineering and formal modeling. Further research will also develop dynamic simulations.

MISSION ENGINEERING

Mission Engineering (ME) is the deliberate planning, analyzing, organizing, and integrating of current and emerging operational and system capabilities to achieve desired warfighting mission effects. The analyses in ME provides information on combat effectiveness, affordability of current and future weapon systems and capabilities, and informs DoD acquisition program investment decisions. It is based on a 10-step process that begins with the prioritization of operational mission areas and ends with continuous management of end-to-end mission threads to maintain the execution health of particular capabilities. The effects/kill chains identify operational needs based on the way the forces plan to fight through mission threads captured in the Combatant Commander's operational plans and contingency plans. The effects/kill chains also inform the issue of the systems needed to accomplish a mission within the System of Systems context.

However, mission system methods are needed that trade requirements and design decisions based on evaluation of hazard/risk, cost, and threat adversary properties. The dependencies between cybersecurity and system safety are deficiently understood and methods and tools to assess these dependencies remain immature. In the cybersecurity domain, traditional assurance processes are inadequate, and currently do not tackle the entire end-to-end mission context. Therefore, further development is needed on new metrics, methods, and tools for hazard mitigation as well as application of the previous research to early, conceptual phases.

FORMAL MODELING

As we engineer mission capabilities to meet the dynamic challenges of today's defense environment with comprehensive digital engineering environments, we can identify and address gaps in our SoS architectures supporting critical missions. Formal modeling processes that confront concerns in verification and validation throughout the lifecycle, specifically in the early phases, is deficient. Further research should connect the STPA methodology that currently underpins the MA MBSE Meta-Model to the AADL/Resolute work on formal modeling and validation of computer information flows and software execution to approach the full lifecycle security/safety process. There has been a disconnect between the modeling and the actual software design to fit the system and its functions. Presently, to get a system approved to a certificate of authority in the formal sense within the security domain, there has to be a certification that includes a formal evaluation activity. For software in the DoD, that is running

various sets of software assurance tools. Integrating the Meta-Model with a software assurance tool, or set of tools, and incorporating functional models, will allow the entire chain of the lifecycle to be closed. The different views into the system are helpful for assuring the overall mission is being addressed and can be verified and validated.

An example is if there is a dependence on losses associated with control actions, then the software that generates control actions should be able to autogenerate the code which can be formally provable as being secure. Nevertheless, there is no current capability in the SysML tools to link into a code generation tool to ensure systems assurance. The Meta-Model enables an excellent understanding and grasp on the system architecture in the conceptualization stage, prior to full contract awards.

Thus, the modeling of a system can be done upfront prior to being handed to different contractors and even generate contractual requirements. The modeling also allows dialogue specifically on the desired warfighter operational mission outcomes, and if done with a formal model, the precise statement of components and the relationships among them can be verified and validated for selection into the system. The research will seek to instill methods, processes, and tools that will aid in whether the requirements are the right argument, and if the modeling being done can determine if the right functional descriptions are being built.

DYNAMIC SIMULATIONS

To make risk-informed decisions about where [not] to place security or resiliency solutions, behavioral models are needed to reason about effects of cyber-attacks, weaknesses, and vulnerabilities. Functional and behavioral models have been used in the formal methods community, and these models are used to verify the reachable states of the software or hardware, real-time hardness, and other safety-related properties. However, in the context of cyber physical systems, it is necessary to also formally reason about the *effects on physical systems*. Fortunately, there is robust body of work that addresses this issue, for example control-theoretic models that, like traditional software formal methods, are also able to reason about reachability (physically reachable states), stability, robustness, and performance. In addition, many commercial grade simulation tools are built upon these techniques and formalisms.

However, much – if not all – of the dynamic simulation capabilities take specifications as a given (e.g. analyzing reference tracking performance); these tools are agnostic to where or how specifications are derived to begin with. Existing tools also tend not to speak to the multi-objective problems that are inherent in mission engineering applications.

Therefore, we need linkages between the attributes and artifacts of mission engineering and those of [cyber, physical] systems modeling and analysis. AADL provides one avenue for the former [cyber], while MathWorks Simulink and others provide the latter [physical]. Furthermore, to be a viable tool for the defense community, and to support automated analyses that scale, these must be *semantically strong, formal* linkages.

Previous research tasks within SERC show promise to creating more formalized approaches that incorporate dynamic simulation capabilities. Vetting and researching the capabilities, opportunities, barriers, and breakthroughs of applying the proven research in early phases of the lifecycle to then propagate through end-to-end would provide an advantageous stance to early transition and adoption Digital Engineering.

CONCLUSIONS

An MBSE approach to security assurance in design, in its broadest sense, can integrate an exceptional wealth of specialized knowledge, skill, and shared thought to create the most optimal blueprint for the mission critical systems. MBSE for security assurance can ingest different perspectives, disciplines, and allows for solutions to be moving targets, not just fixed states, to accommodate an everchanging operational environment. While there is still further research to be done, ART-004 established that the Mission Aware MBSE Meta-Model, when applied to a case study, can produce safety and security analyses for tradeoff evaluation and decision-making in design. It also further provided evidence that functional models and formal descriptions, supplemented with attack graphs and attack patterns, may provide an effective path for early stage design and analysis of security for cyber-physical systems. The research also addressed the dependencies between cybersecurity and system safety, enabling them to be more understood than before, and illustrated the gaps and challenges within the traditional assurance community for cybersecurity.

Current research on CPS is still siloed into subdisciplines, causing digital representations to either represent the cyber or the physical part well, but not both. However, due to the increasing complexity of CPS for defense, new MBSE foundations, theories, and tools are needed to design, analyze, and verify these systems at various levels of abstraction. This includes the structural and functional models as well as threat functional models to obtain scalable graph structures for systems analysis.

The research focused on capturing all four modeling goals in a consistent environment using Model-Based Systems Engineering (MBSE) methods, processes, and tools. A primary outcome of this research is the development and maturation of a meta-model capturing primary concepts of system (operations, function, structure, requirements), assurance (loss, loss effect, and loss scenario), and resilience design (functions to avoid, withstand, recover, and adapt) into MBSE tool constructs.

PROJECT TIMELINE & TRANSITION PLAN

This project was conducted as Phase I of an envisioned multi-year research program aiming to produce systems engineering MPTs to enable evaluation of cost and technical risks and opportunity decisions for security assurance in design. The project showed success in feasibility for MPTs that can be used broadly by the community to improve security assurance case analysis

and decision making in system development. The project has documented the recommended combined safety/security assurance methodology and will continue in a second phase to expand the framework and address mission engineering, formal modeling, and dynamic simulations.

APPENDIX A: MISSION AWARE MBSE META-MODEL AND CSRM

The Mission Aware (MA) MBSE Meta-Model stemmed from consolidating research ideas that have been proposed and accomplished through resilience work started within the SERC in 2012. The MA Meta Model is formalized in GraphQL, thereby enabling tool vendors to incorporate Mission Aware within their toolset which in turn enables transition to use for military and other large systems. The MA model leverages MBSE, along with STPA, while bringing the concepts of resiliency and sentinels, to protect against vulnerabilities, into a unified model. This project was a case study to demonstrate a real system usage of the MA Meta Model in order to demonstrate the potential for transition.

VITECH, who has been doing MBSE for years, has created an evolved representation of a meta model for Systems Engineering. It is a representation of critical systems engineering concepts and their interrelationships spanning requirements, behavior, architecture, and test. The model is hierarchical, using self-associations, as a mechanism to manage system complexity. Having a Meta Model of core Systems Engineering concepts with precise semantics is a key requirement defined by the Object Management Group (OMG) for the second version of SysML. Therefore, as we continue to develop the MBSE Meta Model with VITECH and as they work with OMG on the specification of SysML v2, the MA Meta Model will evolve to the standard when SysML v2 is released.

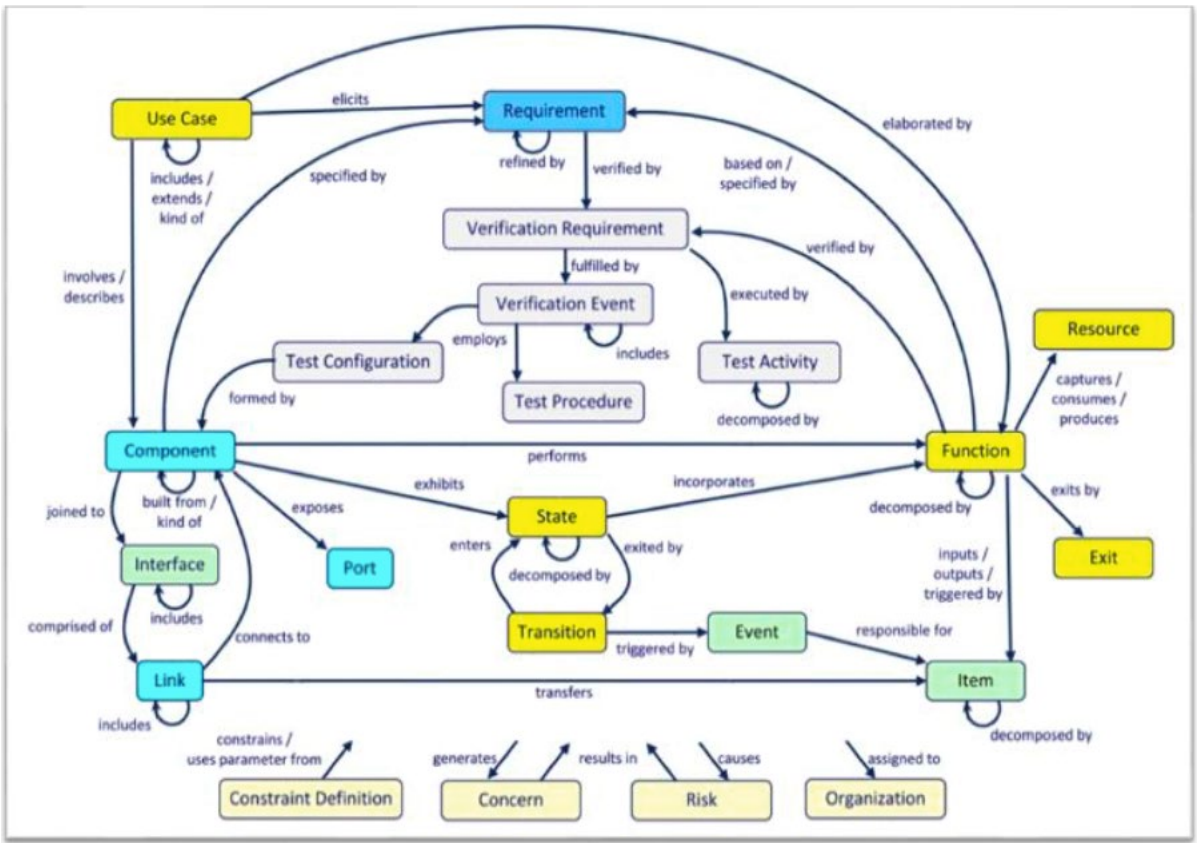

Figure 25. Model-Based Systems Engineering Meta-Model

The MA Meta-Model incorporates STPA to capture losses, hazards, unsafe actions and loss scenarios. As previously discussed, STPA is an iterative, methodical hazard analysis technique to identify causes of hazardous conditions. Within the CPS domain, security can be treated as analogous to safety. The below figure shows the key entities and associations for STPA.

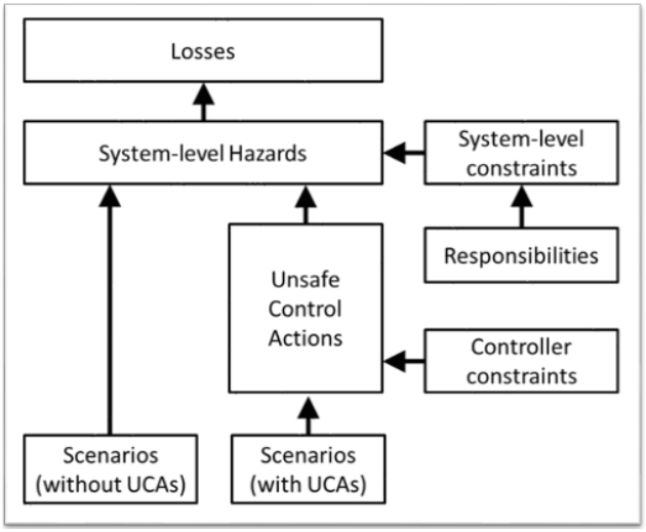

Figure 26. STPA Steps from Handbook

In the Mission Aware work, there is a Resilient Mode with a distinct and separate method of operation of a component, device, or system based upon a diverse redundancy pattern. A Sentinel is another pattern responsible for monitoring and reconfiguration of a system using available Resilient Modes. The Sentinel subsystem is expected to be far more secure than the system being addressed for resiliency.

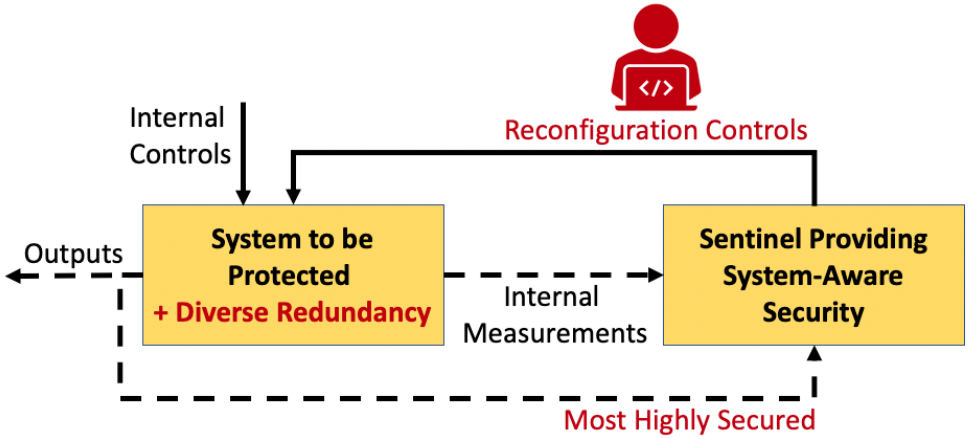

Figure 27. Mission Aware Concepts

The Sentinel monitors a system for illogical behavior, either by monitoring a link of the system, system resource, functions of the system, or even external sensors as demonstrated in the unmanned aerial vehicle sentinel monitoring done at University of Virginia. It is an operational key component, and is not monitoring for cyber-attacks, per se, but monitoring the physical systems to determine what is impossible as well as not normal operations. It may not be a cyber-attack but could be. It is to determine what is happening, and then helps provide diagnosis of why it is happening. For example, if there is an open valve to enable the fullest flow of crude oil within the pipeline, but the reading is saying otherwise, the sentinel would indicate the unusual

output to the normative. It could be a maintenance problem, equipment lifecycle problem, or a cyberattack of some sort that is altering the intended output.

There are countermeasure patterns identified for the Resilient Modes and Sentinels. The following table shows the different patterns, their description, and the attack model countered.

Table 1. Countermeasure Patterns and Attack Model Countered

Counter-measure	*Description*	*Attack model countered*
Trusted Kernel or Guard	Creates a small control system within the CPS that independently monitors and/or manages all resource access	Escalation, interruption attacks
Isolation	Creates an isolated runtime environment (sandbox) for the critical asset that is resistant against attacks.	Escalation, interruption attacks
Redundancy	Replicates the functionality of the critical asset in order to create multiple paths for high availability and fault tolerance in the case of individual function failures	Attacks that disable individual instances of critical assets and functionality.
Diversification	Produces functionally equivalent variations of binaries running in software critical assets. This is an enhancement of the redundancy countermeasure.	Coordinated attacks, zero-day attacks effective in identical binary copies of the critical assets.
Physically Unclonable Function	Secures the integrity and privacy of the messages in the system using a Physical Unclonable Function (PUF) that is hard to predict and duplicate.	Attacks that hijack the communication channels such as man-in-the-middle attacks.
Obfuscation	Obscures the real meaning of data/signals/flows by making them difficult for an attacker to understand. It can use random sources of noise from the environment of the critical assets to increase the entropy.	Attacks that require knowledge of the inner workings of the system, its functions, and its mission.
Parameter Assurance	Compares input data to a table of values in the system to check for large, unexpected deviations.	Attacks that manipulate data files or messages that are sent to the system.
Data Consistency Checking	Verifies the source of a parameter change.	Attacks that use operator specific data entry.
Limiting Circuits	Limits resource use (power, memory) to prevent overload	Power System Attack

The sentinel should be more secure than the system it is monitoring and be very simple itself. Keeping to between 200 and 300 lines of code is ideal and can boast both passive and offensive capabilities. The reconfiguration controls can vary depending on the operational environment(s) of the system. For instance, some armed services can never have an automatic system (e.g., pilots), while others can never be manual but has to be automatic. This is something at design time that is decided and can be amended throughout the lifecycle. The MA-MBSE Meta-Model enables the conversation and provides documentation for decisions made.

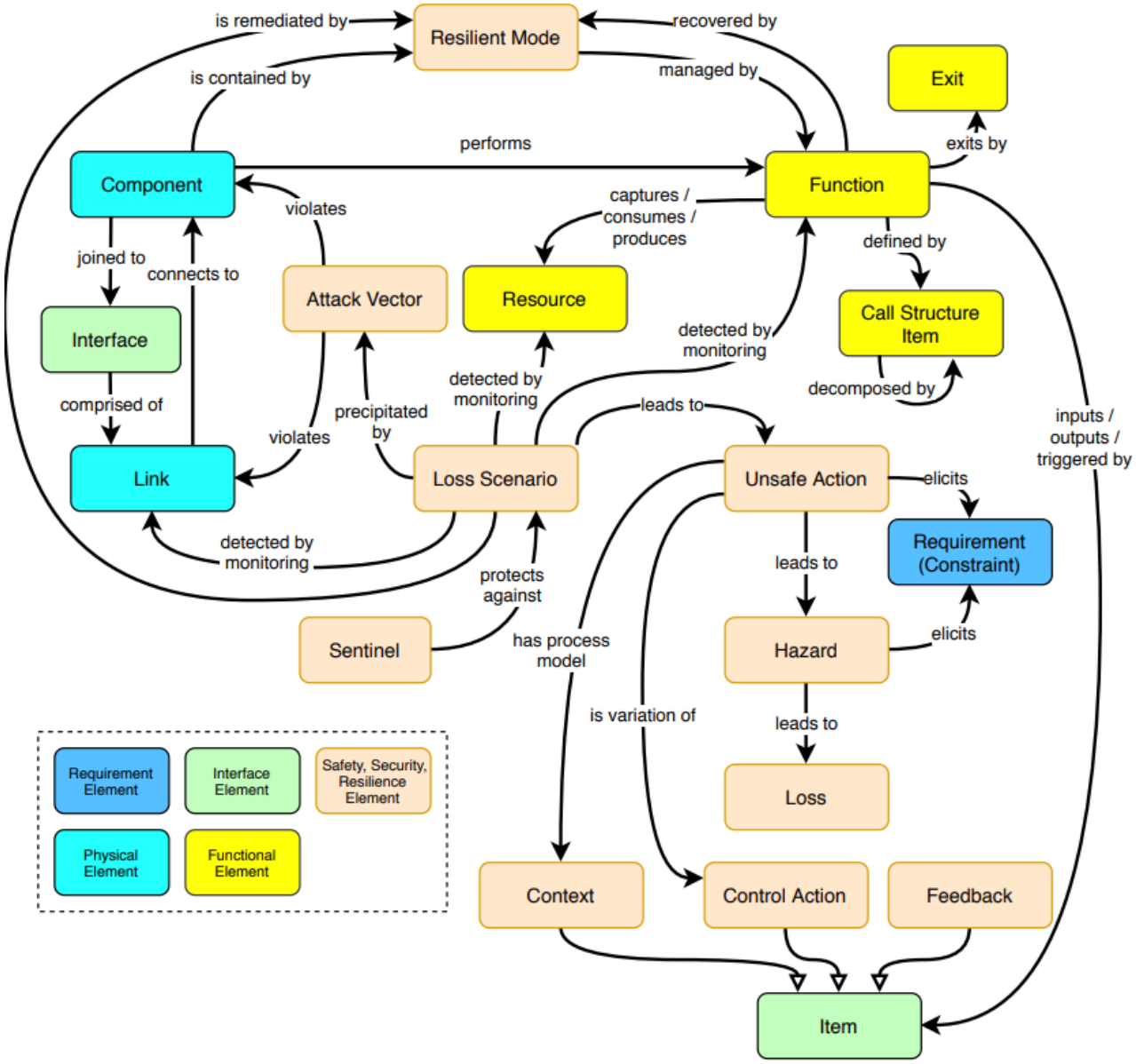

Figure 28. Mission Aware MBSE Meta-Model

The MA-MBSE Meta-Model was produced with the CSRM steps within the model as well. It hosts the following:

1. System Description
    - Component, Link
    - Function, Exit, Resource, Call Structure, Control Action, Feedback, Context
2. Risk Analysis
    - Loss, Hazard, Unsafe Action
3. Resilience Solutions
    - Resilient Mode
4. Cyber Vulnerability Assessment
    - Loss Scenario, Attack Vector

5. Iterate Resilience Solutions (Metrics)
6. Iterate Vulnerability Assessment

CYBER-SECURITY REQUIREMENTS METHODOLOGY (CSRM)

The Cyber Security Requirements Methodology (CSRM) shows what to protect and why, which combination of design patterns to employ in which mission subsystems. There are three entities present with a methodology for evaluating resilience with models – blue team (mission), yellow team (systems engineering team), and red team (threat)(Figure 30). There are multiple threat patterns that can then be included in the model, some of which are listed in the Table 1. Countermeasure Patterns and Attack Model Countered. within the document. The Mission Aware Model-Based System Engineering Meta Model incorporates STPA, MBSE (OMG, SysML), and the Mission Aware Work developed by the UVA through the SERC. The figure below, while shown more linear, is actually comprised of feedback loops. However, most safety and security operations within industry are done in a more linearly, where the blue, yellow, and red teams are approached in series rather than in parallel. It also implies within the model that you can delve into an area at whichever appropriate level of detail is needed. For instance, for the blue team to evaluate risks and the read team to look for vulnerabilities, a level of detail is needed to capture the distinctions.

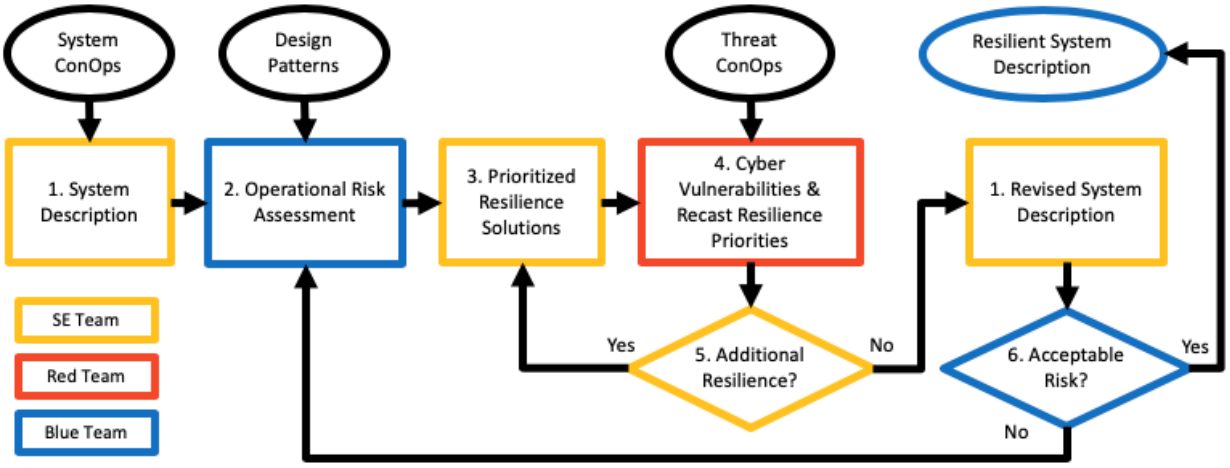

Figure 29. CSRM Flow Chart

APPENDIX B: CONOPS TABLE OF CONTENTS

APPENDIX C: CITED AND RELATED REFERENCES

[1] R. a. McEvilley, "Leveraging System Safety to Improve System Security," in *21st Annual National Defense Industries Association (NDIA) Systems and Mission Engineering Conference*, 2018.

[2] A. M. Madni and S. Jackson, "Towards a Conceptual Frameowrk for Resilience Engineering," *IEEE Systems Journal ,* vol. 3, no. 2, pp. 181-191, 2009.

[3] M. Reed and M. McEvilley, "A Loss-Driven Approach to Systems Analysis," in *22nd Annual NDIA Systems and Mission Engineering Conference*, Tampa, 2019.

[4] P. Beling, B. Horowitz, C. Fleming, S. Adams and B. Georgios, "Model-based engineering for funcitonal risk assessment and design of cyber resilient systems," Charlottesville, 2019.

[5] B. Horowitz, P. Beling, C. Fleming and S. Adams, "Security engineering FY17 Systems Aware Cybersecurity," Systems Engineering Research Center, Hoboken, 2017.

[6] B. Horowitz, P. Beling, C. Fleming, S. Adams and T. Sherburne, "Cyber security requirements methodology (No. SERC-2018-TR-110)," Systems Engineering Research Center, Hoboken, 2018.

[7] B. Horowitz and P. Beling, "Secuirty Engineering Project (No. SERC-2015-TR-036-4)," Systems Engineering Research Center, Hoboken, 2015.

[8] R. Wei, T. P. Kelly, X. Dai, S. Zhao and R. Hawkins, "Model based system assurance using the structured assurance case metamodel," *Journal of Systems and Software,* vol. 154, no. https://doi.org/10.1016/j.jss.2019.05.013, pp. 211-233, August 2019.

[9] "Information Technology, System Security Engineer," 2020. [Online]. Available: http://acqnotes.com/acqnote/careerfields/system-security-engineer.

[10] ISO/IEC/IEEE, "ISO/IEC/IEEE 15026-1:2019(en) Systems and software engineering — Systems and software assurance," 2020. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:15026:-1:ed-1:v1:en.

[11] T. Kelly, "Are 'safety cases' working?," *Safety Critical Systems Club Newsletter,* vol. 17, no. 2, pp. 31-33, 2008.

[12] Object Management Group, "Structured Assurance Case Metamodel (SACM) Version 2.1," OMG Stuctured Assurance Case Model, Inc. , Needham, MA, 2019.

[13] IEEE, "IEEE Standard - Adoption of ISO/IEC 15026-2:2011 Systems and Software Engineering - Systems and Software Assurance Part 2: Assurance Case," IEEE Computer Society, New York, NY, 2011.

[14] B. Boehm, J. Lane, T. Ender, D. Rhodes, A. Ross, K. Sullivan, G. Witus, R. Peak and R. Madachy, "Tradespace and Affordability - Phase 1 (SERC-2013-TR-039-1)," Systems Engineering Research Center, Hoboken, NJ, 2013.

[15] B. Boehm, J. Lane, T. Ender, D. Rhodes, A. Ross, K. Sullivan, G. Witus, R. Peak and R. Madachy, "Tradespace and Affordibility - Phase 2 (SERC-2013-TR-039-2)," Systems Engineering Research Center, Hoboken, NJ, 2013.

[16] B. Boehm, J. Lane, T. Ender, D. Rhodes, A. Ross, K. Sullivan, G. Witus, R. Peak, R. Madachy and M. Yukish, "System Qualities Ontology, Tradespace and Affordability (SQOTA) Project – Phase 4 (SERC-2016-TR-101)," Systems Engineering Research Center, Hoboken, NJ, 2016.

[17] B. Boehm, T. Ender, D. Jacques, J. Lane, R. Madachy, A. Ross, K. Sullivan, G. Witus and M. Yukish, "-ilities Tradespace and Affordability Project – Phase 3 (SERC-2014-TR-039-3)," Systems Engineering Research Center (SERC), Hoboken, NJ, 2014.

[18] B. Boehm, T. Ender, D. Jacques, J. Lane, R. Madachy, D. Rhodes, A. Ross, K. Sullivan, G. Witus and M. Yukish, "System Qualities Ontology, Tradespace and Affordability (SQOTA) Project Phase 5 (SERC-2017-TR-105)," Systems Engineering Research Center (SERC), Hoboken, NJ, 2017.

[19] B. Boehm, T. Ender, D. Rhodes, K. Sullivan, G. Witus and M. Yukish, "System Qualities (SQs) Ontology, Tradespace and Affordability (SQOTA), Phase 6 (SERC-2018-TR-108)," Systems Engineering Research Center (SERC), Hoboken, NJ, 2018.

[20] B. Boehm, "System Qualities Ontology, Tradespace, and Affordability (SQOTA) Phases 1-7 (SERC-2019-TR-012)," Systems Engineering Research Center (SERC), Hoboken, NJ, 2019.

[21] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau and R. McQuaid, "Developing Cyber Resilient Systems: A Systems Security Engineering Approach (NIST Special Publication 800-160 Volume 2)," NIST, Gaithersburg, MD, 2019.

[22] R. Ross, M. McEvilley and J. C. Oren, "Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems, NIST Special Publication 800-160 VOLUME 1," NIST, Gaithersburg, MD, 2016.

[23] B. I. Rachowitz, R. K. Maue, N. P. Angrisano and B. Abramson, "A guide to engineering workstations: using workstations efficiently," *IEEE Spectrum,* vol. 28, no. 4, pp. 38-40, 1991.

[24] C. Alberts, R. J. Ellison and C. Woody, "Cyber Assurance: CERT Research Report," Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 2009.

[25] SEBoK, "SEBoK: Guide to the Systems Engineering Body of Knowledge," INCOSE, Stevens Institute of Technology, IEEE Systems Council, 15 May 2020. [Online]. Available: https://www.sebokwiki.org/wiki/System_Assurance_(glossary).

[26] T. A. McDermott, M. M. Clifford, C. Arquimedes, G. Quirós and V. B. Sitterle, "System Assurance in the Design of Resilient Cyber-Physical Systems," in *Design Automation of Cyber-Physical Systems*, New York, NJ, Springer, 2019.

[27] D. M. Nicole, W. H. Sanders and K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Transactions on Dependable and Secure Computing,* vol. 1, no. 1, pp. 48-65, 2004.

[28] W. Young and N. G. Leveson, "Systems thinking for safety and security," in *Annual Computer Security Applications Conference*, New Orleans, LA, 2013.

[29] N. Leveson, Engineering a Safer World, Cambridge: MIT Press, 2012.

[30] N. Leveson, "Engineering a safer world: Systems thinking applied to safety," MIT Press, Cambridge, MA, 2011.

[31] N. G. Leveson and W. Young, "An Integrated Approach to Safety and Security Based on Systems Theory," *Communications of the ACM,* vol. 57, no. 2, pp. 31-35, 2014.

[32] N. G. Leveson and J. P. Thomas, "STPA Handbook," March 2018. [Online]. Available: http://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf.

[33] K. Fisher, J. Launchbury and R. Richards, "The HACMS Program: using formal methods to eliminate exploitable bugs," *Phil. Trans. R. Soc.,* 04 September 2017.

[34] DARPA, "High-Assurance Cyber Military Systems (HACMS)," [Online]. Available: https://www.darpa.mil/program/high-assurance-cyber-military-systems.

[35] R. Richards, "Cyber Assured Systems Engineering (CASE)," 2020. [Online]. Available: https://www.darpa.mil/program/cyber-assured-systems-engineering.

[36] J. Bergenthal and M&S Committee, "Final Report: Model Based Engineering (MBE) Subcommittee," NDIA, 2011.

[37] INCOSE Technical Operations, "Systems Engineering Vision 2020," INCOSE, Seattle, WA, 2007.

[38] S. L. Cornford and M. S. Feather, "Model Based Mission Assurance in a Model Base Systems Engineering (MBSE) Framework," NASA, Hampton, VA, 2016.

[39] P. Zimmerman, "Digital Engineering Strategy and Implementation," in *NIST Model-Based Enterprise Summit*, Gaithersburg, MD, 2019.

[40] Department of Defense , "DoD Digital Modernization Strategy," Washington, D.C., 2019.

[41] T. McDermott, "A Rigorous System Engineering Process for Resilient Cyber-Physical Systems Design," in *IEEE 2019 International Symposium on Systems Engineering (ISSE)*, Edinburgh, UK, 2019.

[42] P. H. Feiler and D. P. Gluch, Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language, Upper Saddle River, NJ: Addison-Wesley, 2012.

[43] A. Gacek, J. Backes, D. Cofer, K. Slind and M. Whalen, "Resolute: An Assurance Case Language for Architecture Models," *ACM SIGAda Ada Letter,* vol. 34, no. 3, pp. 1094-3641, October 2014.

[44] Y. Wadhawan and C. Neuman, "Evaluating Resilience of Gas Pipeline Under Cyber-Physical Attacks: A Function-Based Methodology," in *CPS-SPC*, Vienna, Austria, 2016.

[45] National Cybersecurity & Communications Integration Center (NCCIC), "Awareness Briefing: Russian Activity Against Critical Infrastrcture," DoD, 25 July 2018.

[46] R. K. Abercrombire, F. T. Sheldon and M. R. Grimaila, "A Systematic Comprehensive Computational Model for Stake Estimation in Mission Assurance - Applying Cyber Security Econometrics System (CSES) to Mission Assurance Analysis Protocol (MAAP)," in *IEEE Second International Conference on Social Computing*, Minneapolis, MN, 2010.

[47] D. J. Bodeau and R. Graubart, "Cyber Resiliency Engineering Framework," MITRE, Bedford, MA, 2011.

[48] M. R. G. F. T. S. Robert K. Abercrombie, "A Systematic Comprehensive Computational Model for Stake Estimation in Mission Assurance - Applying Cyber Security Econometrics System (CSES) to Mission Assurance Analysis Protocol (MAAP)," in *IEEE International Conference on Social Computing*, 2010.

[49] J. Aguilar, "Design Assurance Guide," 4 June 2009. [Online]. Available: aerospace.wpengine.netdna-cdn.com/wp-content/uploads/2015/04/TOR-20090591-11-Deisgn-Assurance-Guide.pdf..

[50] Y. W. J. T. J. J. P. ,. a. B. K. Deng, "An approach for modeling and analysis of security system architectures ," *IEEE Transactions on Knowledge and Data Engineering,* pp. 1099-1119, 2003.

[51] G. T. I. A. C. Syllabus, *INTA 4014/6014: Scenario writing and path gaming,* Atlanta, GA: Georgia Tech, 2019.

[52] B. a. F. Horowitz, "Technical Report SERC-2017-TR-114, Security Engineering – FY17 Systems Aware Cybersecurity," SERC, Hoboken , 2017.

[53] M. a. Yang, "Workshop Report: SERC-2017-WR-101: Model-Based System Assurance: Enabled by Digital Engineering," SERC, Hoboken, 2017.

[54] P. a. Davarynejad, "Calculating Adversarial Risk from Attack Trees: Control Strength and Probabilistic Attackers," Springer International Publishing , 2015.

[55] J. Rasmussen, Information Processing and Human Machine Interaction: An Approach to Cognitive Engineering, North-Holland, 1986.

[56] K. Vicente, Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work, CRC Press, 1999.

[57] Wortman, Tehranipoor and a. Chandy, "An Adversarial Risk-based Approach for Network Architecture Security Modeling and Design," in *18th Annual Conference on Systems Engineering Research*, Charlottesville, VA, 2018.

[58] W. Zeng, "A methodology for cost-benefit analysis of information security technologies," *Concurrency and Computation Practice and Experience,* vol. 31, no. 4, 2018.