Cyber Security Requirements Methodology

Technical Report SERC-2018-TR-110

July 26, 2018

Principal Investigator:

Dr. Barry Horowitz, University of Virginia

Co-Principal Investigator:

Dr. Peter Beling, University of Virginia
Dr. Cody Fleming, University of Virginia

Research Team:

Stephen Adams, University of Virginia
Bryan Carter, University of Virginia
Tim Sherburne, University of Virginia
Carl Elks, Virginia Commonwealth University
Georgios Bakirtzis, Virginia Commonwealth University
Forrest Shull, Software Engineering Institute
Nancy R. Mead, Software Engineering Institute

Sponsor: DASD(SE)

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 HQ0034-13-D-004 (Task Order 0538).

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, SEQUENCES)

EXECUTIVE SUMMARY

This report addresses the DoD/Army/SERC-sponsored, UVA-led 9 month research effort to develop a methodology for establishing cyber security requirements at the preliminary design phase of new physical systems programs. The requirements addressed include the integration of cyber attack defense and resilience solutions, as well as security-related software engineering solutions. Referred to as Cyber Security Requirements Methodology (CSRM), the developed process includes six sequential steps conducted by three teams (an operationally focused team, a cybersecurity focused team and a systems engineering team). Model-based engineering tools were utilized to support each of the steps. A trial weapon system use case was conducted to gain an initial evaluation of the methodology. The use case system, referred to as Silverfish, was hypothetical, but deemed as a reasonable representation of a possible weapon system. Results of the trial were promising and point to a number of possible paths for follow-on research including implementing the methodology on a real system and building the necessary tools to scale up the methodology to a real system.

1. INTRODUCTION

Historically, both the cyber security and system engineering (SE) communities have pointed to the desirability for addressing cyber security requirements early in the overall design process for new systems. Prior University of Virginia (UVA) research efforts, referred to as System Aware Cyber Security, have addressed cyber attack resilience requirements as a subject associated with the design of cyber physical systems. Correspondingly, one would expect to address cyber attack resiliency early on in an organization's processes for system design. As related to the topic of this paper, the UVA research team has recognized that cyber attack resilience requirements need to be considered in the context of other aspects of cyber security (e.g., cyber security defense requirements, software quality management as related to cyber security) because the different mechanisms for addressing system cyber security can serve to efficiently complement each other in achieving an overall desired level of protection. The recognition of the desirability to consider the multiple aspects of cyber security concurrently in order to properly address resiliency requirements served to motivate the UVA research team to develop an integrated cyber security requirements methodology. A multi-organization research team was formed for defining the methodology, consisting of UVA (team lead), the Software Engineering Institute (SEI), the Virginia Commonwealth University (VCU) and the US Army's Armament Research Development and Engineering Center (ARDEC). Each of these organizations brought a particular focus required by the research activity; UVA/SE, SEI/cyber attack threat analysis, VCU/cyber attack analysis tool development, ARDEC/ weapon system design. The team defined a 9-month project to develop a cyber security requirements methodology (referred to as CSRM) that could be embedded within the preliminary design timelines used for DoD development projects and a first trial application to serve as a basis for evaluation and refinements to the methodology. Section 2 outlines the resulting methodology. Section 3 describes the hypothetical weapon system used as the initial trial use case. Section 4 highlights a set of analysis and prototyping/simulation tools developed to support the CSRM. Section 5 presents results from the trials. Section 6 provides an assessment of results derived from the project and areas where future research can contribute to advancing the opportunity for addressing cyber security requirements for cyber physical systems.

In order to suitably address the cyber attack resilience aspect of cyber security, the UVA SE team developed the following definition related to cyber physical systems as a derivative of a broader resilience definition presented by the Idaho National Laboratory in 2009:

> *Cyber Attack Resilience* - the capacity of a system to maintain state awareness (implies a monitoring process of physical and software-related states) as a means for detecting cyber attacks, and to proactively maintain a safe level of operational normalcy through rapid system reconfigurations in response to detected cyber attacks that would impact system performance. Maintaining operational normalcy includes containing the immediate consequences of the detected attack and post-attack forensic support based upon the data collected for detecting attacks.

As part of UVA's System Aware Cyber Security concept, the required anticipatory processes for monitoring and reconfiguration is conducted by a subsystem referred to as a *Sentinel*, which should be far more secure than the system being addressed for resiliency. While the cyber attack detection process is expected to be automated, the level of reconfiguration automation may vary across system functions:

- Totally Automated (Sentinel determines what to do and informs appropriately trained system operators regarding automated execution)
- Semi-automated (System operators receive automated recommendation(s) from Sentinel and, accounting for both battle context and a broader set of information available to them, decide on what to do)
- Manual (Operators, or higher levels in the command hierarchy, determine what to do)

2. CYBER SECURITY REQUIREMENTS METHODOLOGY (CSRM) DESCRIPTION

This section presents the six-step cyber security requirements (CSRM) methodology that would be carried out by three collaborating teams, derived as a result of the research efforts discussed in this paper. In addition, it introduces how analysis and rapid prototyping/simulation tools can be used to support decision-making regarding cyber security requirements. Section 4 discusses the use of the SysML and analysis tools in detail.

The CSRM for cyber physical systems introduced in this research activity is risk-based. Risk is determined by the consequences that would occur should a particular cyber attack scenario occur and the likelihood of that scenario actually occurring. Consequences can range, for example, from human injury or loss of life, to loss of control, to corruption or delays of situation awareness information, to denial of a system operation. The CSRM recognizes that the owners, operators and users of a system are the appropriate community of people to consider and prioritize the potential consequences that need to be avoided. The CSRM also recognizes that the cyber attackers (adversaries) are the community of people that prioritize and ultimately determine the likelihood of specific cyber attacks occurring. Cyber security solutions are intended to influence the likelihood of attacks and, in particular, cyber attack resiliency solutions are intended to address the consequences of detected attacks. The six-step CSRM is divided in a manner that addresses this division of risk and the three teams that execute the CSRM provide the knowledge required to address the six steps.

The objective of CSRM is to augment current preliminary design efforts for new cyber physical systems with a timely and efficient process that addresses the cyber security requirements for the system. As discussed in Section 1, the individual elements for achieving cyber security (e.g., cyber attack defense, cyber attack resilience) are complementary, and would best be done in a collective effort when the new system is being designed. During this phase of system design, important initial decisions can be made regarding system architecture, including for example:

- Separation and isolation of hardware and software supporting different system functions,

- Use and selection of off-the-shelf products, accounting for historical cyber attacks,
- Dependence on defense capabilities, with specific solutions to be selected when design is sufficiently mature,
- Where within the new system's development process to focus the most emphasis and corresponding resources regarding SW development processes (quality assurance tools, testing, developer skills, life cycle support, etc.),
- Design and performance requirements for resilience-related capabilities both for immediate implementation and to facilitate simpler addition in preparation for higher likelihood requirements over the life-cycle,
- Addressing the operator related aspects of resiliency through rapid prototyping experiments and exercise-related support tools.

The six-step CSRM emerged from this research effort as an efficient and potentially high-value mechanism to conduct a risk assessment that would lead to the desired architectural design decisions. The individual steps are listed below:

- Step 1 – High level, tool-based, system description produced by SE, including system architecture and functional description – MagicDraw's SysML implementation was the chosen system description support tool that was used across all 6 steps
- Step 2 – Blue Team consequence analysis, resulting in a prioritized list of system functional problems to be avoided
- Step 3 – SE team derivation of resilience solutions (described via use of SysML) that respond to Blue Team results
- Step 4 – Red Team, based upon experience with cyber attack threats, COTS and GOTS cyber defense solutions and defense and use of a VCU analytical tools for confirmation of attack-related assumptions (discussed in Section 4), prioritizes software engineering solutions, cyber defense solutions and resilience solutions
- Step 5 – SE team adjusts SysML system description to account for Red Team recommendations and rapid prototyping/simulation results for presentation to Blue Team; Initiates cost analysis effort
- Step 6 – Blue Team responds to Red Team recommendations and simulation results with their revised consequence prioritization of solutions, thereby enabling SE team to provide an integrated system design discussion for requirements-related decision-makers that would include considerations of cost, as well as risk reduction.

The CSRM requires three teams to carry out the steps; a systems engineering team (SE Team), a Blue Team and a Red Team. The roles of each of the three teams are presented below.

The SE Team (UVA for the trial use case) would consist of a group of people with a broad range of skills, including technical and operationally related experience. They would be required to have strong analytical skills and the ability to use system description and assessment tools. The team would be required to develop (or provide from the overall cyber physical system project's SE team) an initial high level System Design, without cyber attack-related resilience features, to start

to work with. Based upon the Blue Team's prioritized consequence avoidance assessment, the SE Team would derive potential resilience features and the architecture for their implementation (e.g., 3 or 4 possible resilience system augmentations for consideration). After receiving the Red Team's prioritized solution assessments, the SE Team would derive integrated solution alternatives that account for the full risk analysis (sensitive to both Blue and Red Team analyses).

The SE Team would also be responsible for the coherent management of the methodology process, updating the system descriptions to account for the new solutions as they emerge from the CSRM process.

The Blue Team (ARDEC for the trial use case) would be an operationally-oriented group, including members experienced in addressing use of systems under duress (not necessarily cyber attacks, but perhaps electronic warfare attacks or weapon-fire attacks). It would be desirable for the Blue Team to have knowledge regarding operational practices, and their purposes, for legacy systems that were related to the system to be developed. The team would focus on the Consequence component of risk, providing a prioritized view for the various system functions of consequences to be avoided (e.g., denial of service, corruption of information to operators, delays in execution, etc.). As required, the Blue Team would be supported by the SE Team regarding interpretation of the tool-based representation of the system under consideration. An important CSRM attribute is that Consequence analysis need not include inputs from cyber security experts.

The Red Team (SEI, VCU for the trial use case) would be focused on the identification of likelihoods of potential cyber attacks, both with and without the application of potential solutions to the overall system design. The team would provide a view on the relative efficacy of different cyber security solutions, prioritizing the relative importance of SW quality solutions, defense solutions and resiliency solutions, including considerations of past cyber attacks and SW vulnerabilities to attack. The members of the team would be expected to pose alternative solutions and assessments of the corresponding impact of potential solutions regarding related cyber attack likelihoods. An important attribute of the CSRM is that the Red Team consists of a mixture of cyber attack expertise and cyber security expertise, working together to iteratively develop an assessment that relates solution selection with likelihoods for influencing attack likelihoods.

The following section of the paper describes the trial use case for the initial evaluation of the CSRM.

3. USE CASE DESCRIPTION (SILVERFISH)

As part of preparing for an initial trial of the CSRM, an initial weapon system (to be referred to as Silverfish) use case was developed to serve as an initial application. Silverfish is a hypothetical system, but was deemed by the ARDEC team as sufficient for the purposes of CSRM development. In addition to supporting the development of the CSRM, the Silverfish use case is also intended

to support research related to both decision support tool development and rapid prototyping/simulation efforts to help identify potential system resilience solutions. Section 3.1 provides a functional description of Silverfish and Section 3.2 provides a corresponding SysML-based description.

3.1 SILVERFISH FUNCTIONAL SYSTEM DESCRIPTION

The Silverfish system is a rapidly deployable set of fifty (50) individual ground-based weapon platforms (referred to as obstacles) controlled by a single operator. The purpose of the system is to deter and prevent adversaries from trespassing into a designated geographic area that is located near a strategically sensitive location. The system includes a variety of sensors to locate and classify potential trespassers as either personnel or vehicles. An internal wireless communication system is used to support communication between the sensors and the operator, and also supports fire control communications between the operator and the obstacles. The sensors include obstacle-based seismic and acoustic sensors, infrared sensors and an unmanned aerial vehicle-based surveillance system to provide warning of potential adversaries approaching the protected area. The operator is located in a vehicle, and operates within visual range of the protected area. The operator is in communication with a higher-level command and control (C2) system for exchange of doctrinal-related and situation awareness information. A more detailed functional description of the system is presented below. Section 3.2 provides the corresponding SysML representation for Silverfish.

- Purpose: Deter and prevent, when and where necessary, via the use of rapidly deployable obstacles, adversarial tracked vehicles (assumed maximum speed - 10mph) or individuals from trespassing into geographic areas that are close to strategically sensitive locations.
- Prohibited Area: ~100 acres of open field space (100 acres, approximately 0.16 square miles = 0.4 mile x 0.4 mile area). At maximum speed a vehicle would take about 3 minutes to cross the prohibited area.
- Obstacle Deployment: About 50 obstacles are available to be distributed over the 100-acre protected area (each obstacle is designed to protect a 300x300 foot area). Two types of obstacles can be deployed. One type of obstacle addresses anti-personnel requirements. It contains six (6) short-range sub-munitions, each covering a 60-degree portion of a circular area to be protected. The second type of obstacle contains a single munition capable of impacting a tracked vehicle.
- Operation: The operator, located in a vehicle that is operated close to the prohibited area (~150 meters away), remotely controls individual obstacles and their sub-munitions, based upon sensor-based and operator visual surveillance of the prohibited area.
- Prohibited Area Surveillance: The operator is supported by obstacle-based acoustic and seismic sensors (geophones and accelerometers) that can detect and distinguish between vehicles and people, redundant infrared sensors that can detect and track the movement of people and vehicles, and real-time Video/IR derived early warning information regarding people and vehicles approaching the prohibited area provided by

a UAV managed by the operator. The UAV is used to provide warning information. The operator can relocate his or her vehicle for improved visual observation.

- Obstacle design features: The obstacle-based sensors provide regular operator situation awareness reports (seconds apart) when they detect a trespasser. They provide, at a lower data rate (e.g., a minute apart), general health related information, including reports on their location (GPS-based), their on-off status, and their remaining battery life. Should a weapon be fired, the obstacle confirms the acceptance of commands and the actual firing events. To address potential tampering risks, obstacle-based software can only be modified by electrically disconnecting their platform-based computer from the obstacle, and removal results in self-destruction of that computer.
- Infrared sensor configuration: A single pole-mounted IR sensor is assumed to be capable of providing surveillance of the entire protected area. A second sensor is provided for redundancy, and can be used to provide surveillance of areas that the single sensor is not able to observe. The IR sensors provide the same type of operator situation awareness data at the same rates as the obstacle-based sensors, but in addition provide tracking information to enable the operator to project future locations of moving vehicles or people.
- Requirements for Avoiding Errors: Concerns exist regarding detonating sub-munitions in cases where non-adversarial vehicles or people, by chance, enter the prohibited area. Concerns also exist about failing to fire munitions when an adversary is approaching a strategically sensitive location via the prohibited area. The operator, when possible, can use visual observations to increase confidence regarding fire control.
- Operator Functions: The operator can set the obstacles into either on or off modes and can cause individual or designated groups of obstacles/sub-munitions to detonate when in on mode. Obstacles can be commanded to self-destroy designated critical information in order to prevent adversaries from collecting such information for their own purposes. The operator also can launch a quad-copter drone (UAV) to provide video/IR based early warning information regarding potential trespassers of the protected area (~ 5 minute warning for vehicles approaching at a 10 mph speed).
- Communications Systems: The Operator, the higher level C2 System, and UAV operate on a shared radio system that is integrated to a relay node(s) that couples into the Silverfish system's integrated wireless communication network. The communication system includes digital interfaces that support formatted data transfers between the operator's system, the UAV subsystem, the individual obstacles, the IR subsystem, and the C2 Center. The communication system also supports short message text and voice communications between operator and C2 system.
- Operator Control Station: The operator is provided with a vehicle-mounted computer(s) subsystem that provides situation awareness information including individual obstacle status, and sensor-based situation awareness information. The subsystem also provides computer-based entry and corresponding weapon system feedback for fire control-related inputs from the operator. The control station also supports required digital situation awareness-related reporting to the C2 center, as well as support for UAV control.

- Command Center Controls: The C2 center digitally provides weapon control information for the operator (determines weapon system on/off periods, provides warning of periods of higher likelihood of attack, provides forecasts of possible approach direction to the prohibited area, enables operation with/without UAV support, etc.). As determined by either the operator or the C2 center, out of norm situations can be supported through rapid message communications between the C2 center and the operator.
- Forensics: All subsystems collect and store forensic information for required post-mission analysis purposes.
- Rapid Deployment Support: All subsystems enable rapid deployment testing to confirm readiness for operational use.

3.2 SILVERFISH SYSML-BASED SYSTEM DESCRIPTION

Initial SysML representations were created prior to the Blue Team meeting associated with step 2 of the CSRM. These initial SysML descriptions define the basic composition, architecture, and concept of standard operation for the Silverfish system. The major components of the baseline system are defined in a SysML block definition diagram along with basic functional descriptions of the information exchanged between each component, seen in the Figure 3.1 below.

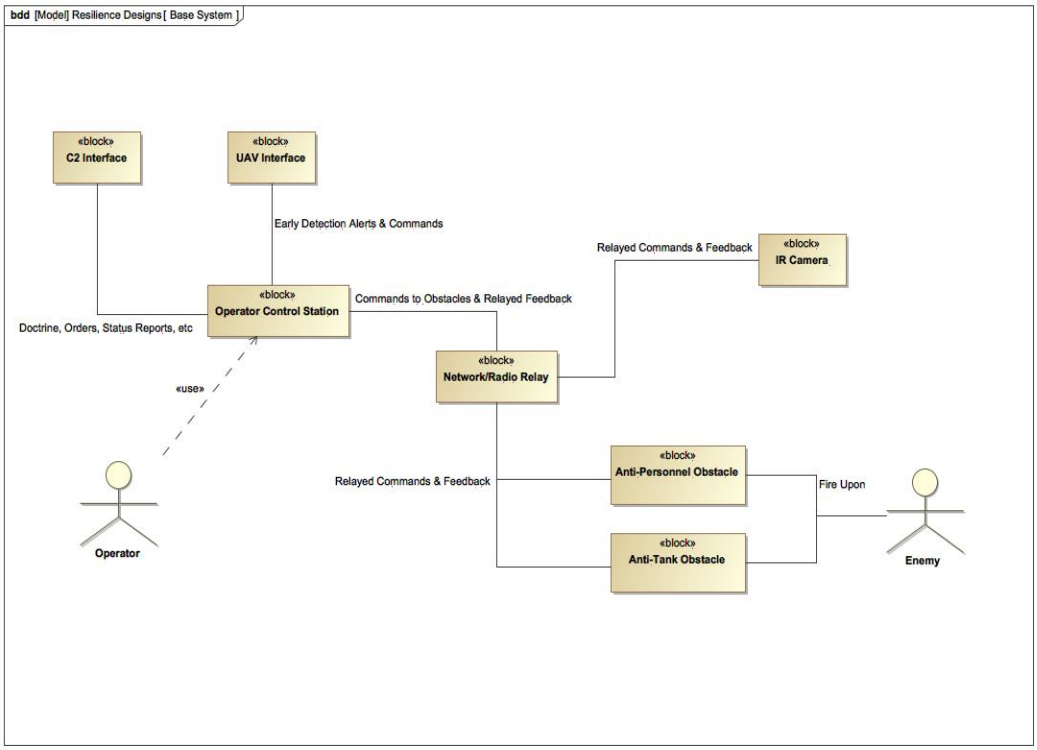

Figure 3.1 - SysML Block Definition Diagram for Baseline Silverfish System Architecture

The block definition diagram (BDD) provides a simple graphical summary of the Silverfish system as described in Section 3.1. More specific details, such as the parameters for field size, sensor range, etc. are embedded in the SysML objects as descriptive information, thus this information is not visible in the standard SysML diagrams.

The SysML internal block diagram (IBD) for the baseline Silverfish system, presented in Figure 3.2, shows a more detailed representation of the parts that make up the larger components presented in the BDD above, as well as defining the hardware type of each main computer (e.g. the operator control station is a computer). Furthermore, the specific functions and traits of each larger component are defined as 'Classifier Behavior' blocks. For instance, in the IBD for the baseline Silverfish system below, the Operator Control Station provides the fire control function and utilizes encryption, and the AP Obstacle also utilizes encryption and is composed of an AP munition, hardware-software interface for controlling its munitions, and an acoustic sensor.

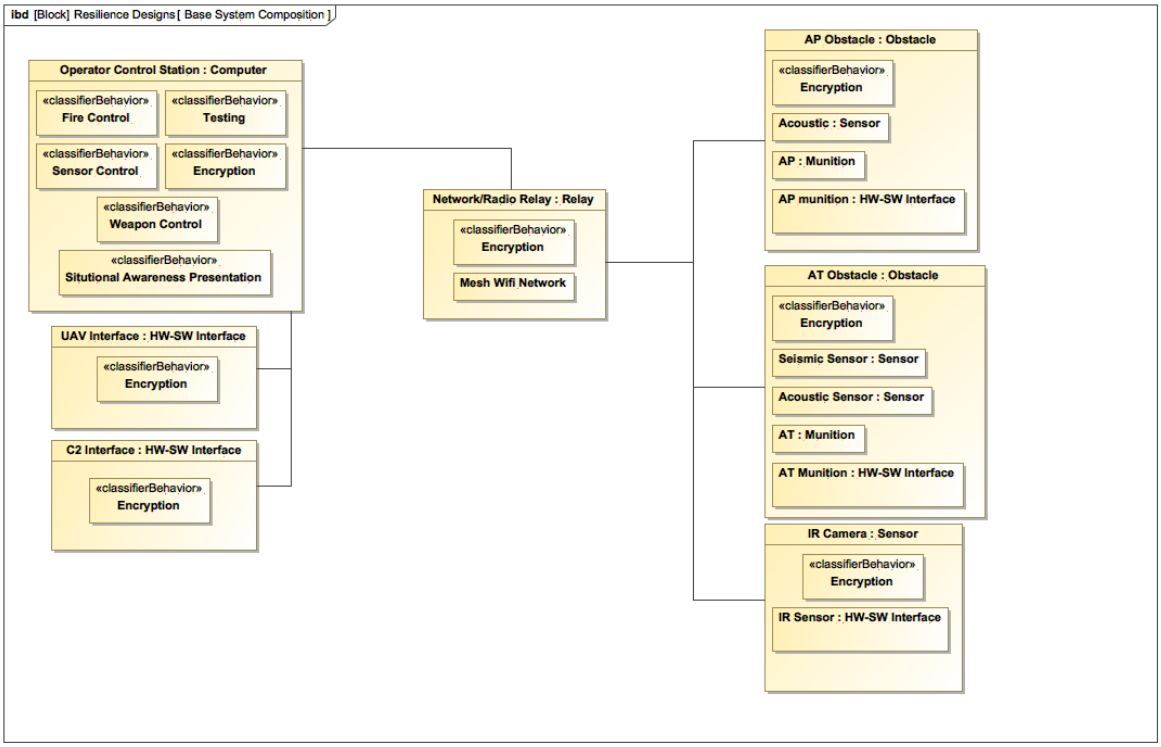

Figure 3.2 - SysML Internal Block Diagram for Baseline Silverfish System Architecture

Finally, the SE team developed a basic graphical representation of the standard operating procedure for using Silverfish in a combat scenario using the SysML activity diagram, presented in Figure 3.3. This diagram outlines the basic functions performed by both the operator and the Silverfish system to engage a target within the denied area.

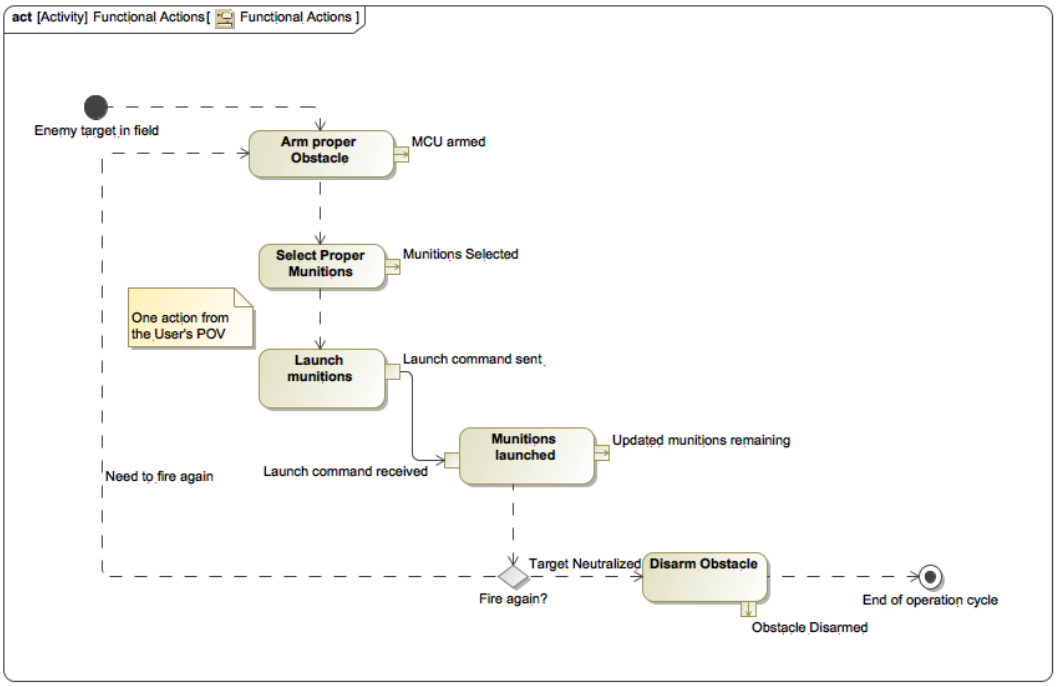

Figure 3.3 - SysML Internal Block Diagram for Baseline Silverfish System Architecture

In this diagram, the large action blocks indicate steps in firing a munition on an enemy, the decision node alludes to the operator's damage assessment duties, and the nodes on each action block represent software and hardware changes to the system to allow firing on an enemy target. The conceptual view of operation presented in this diagram is later used by the SE team as a part of its hazard analysis.

These initial Silverfish SysML representations create a foundation from which the model is updated throughout the CSRM process. Following Step 2 of the CSRM, the undesirable consequences defined by Blue Team are converted into requirements within a SysML requirements diagram. Alternative resilient system compositions are appended to the SysML model in Step 3 prior to the Red Team meeting. Finally, the SysML model is further updated following the Red Team inputs. These additional SysML representations can be found in Section 5 of this report.

4. Development and Application of Rapid Prototyping and Analytical Tools for Establishing System Cybersecurity Requirements

This section will describe the results of research efforts for applying user focused rapid prototyping and simulation and analysis tools to support development of system cyber security requirements. Section 4.1 presents the simulation and prototyping infrastructure developed for this effort and the cyber security-related value that can be produced through application. Section 4.2 presents the application and development of analysis tools to support prioritization of system cyber security requirements.

4.1 Rapid Prototype and Simulation Tools

Rapid Prototyping and Simulation provide a mechanism to explore system resiliency design alternatives and associated user experience impacts early in the development lifecycle before committing to a specific design and implementation.

Details of the Silverfish prototype and simulation are described below utilizing a variety of system description methodologies.

4.1.1 Silverfish Prototype Context Diagram

The Unified Modeling Language (UML) is used to describe the scope of Silverfish prototype and the external interfaces (Actors) to the system using a Context Diagram (shown below).

As described in the previous section, the Silverfish System includes both Fire Control functions (shaded orange) and Situational Aware functions (shaded blue). The Silverfish System supports a single Operator with User Interfaces to track Physical Attackers via the Situational Aware Application and, if needed, to fire upon Physical Attackers via the Fire Control Application. The Sentinel System (shaded green) monitors the Silverfish System for Cyber Attacker threats and is able to re-configure the Silverfish System in order to contain attacks and to provide System Resiliency options for the Operator.

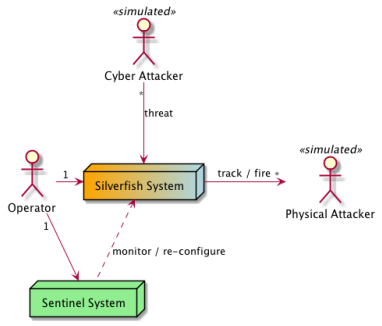

Figure 4.1.4 - Silverfish Context Diagram

4.1.1.2 SILVERFISH PROTOTYPE DATA MODEL

The structural requirements for the Silverfish Prototype are described via a UML Class Diagram (shown below). The diagram describes the system objects, their attributes, operations (or methods), and the relationships among objects. A table of descriptions is provided below the diagram.

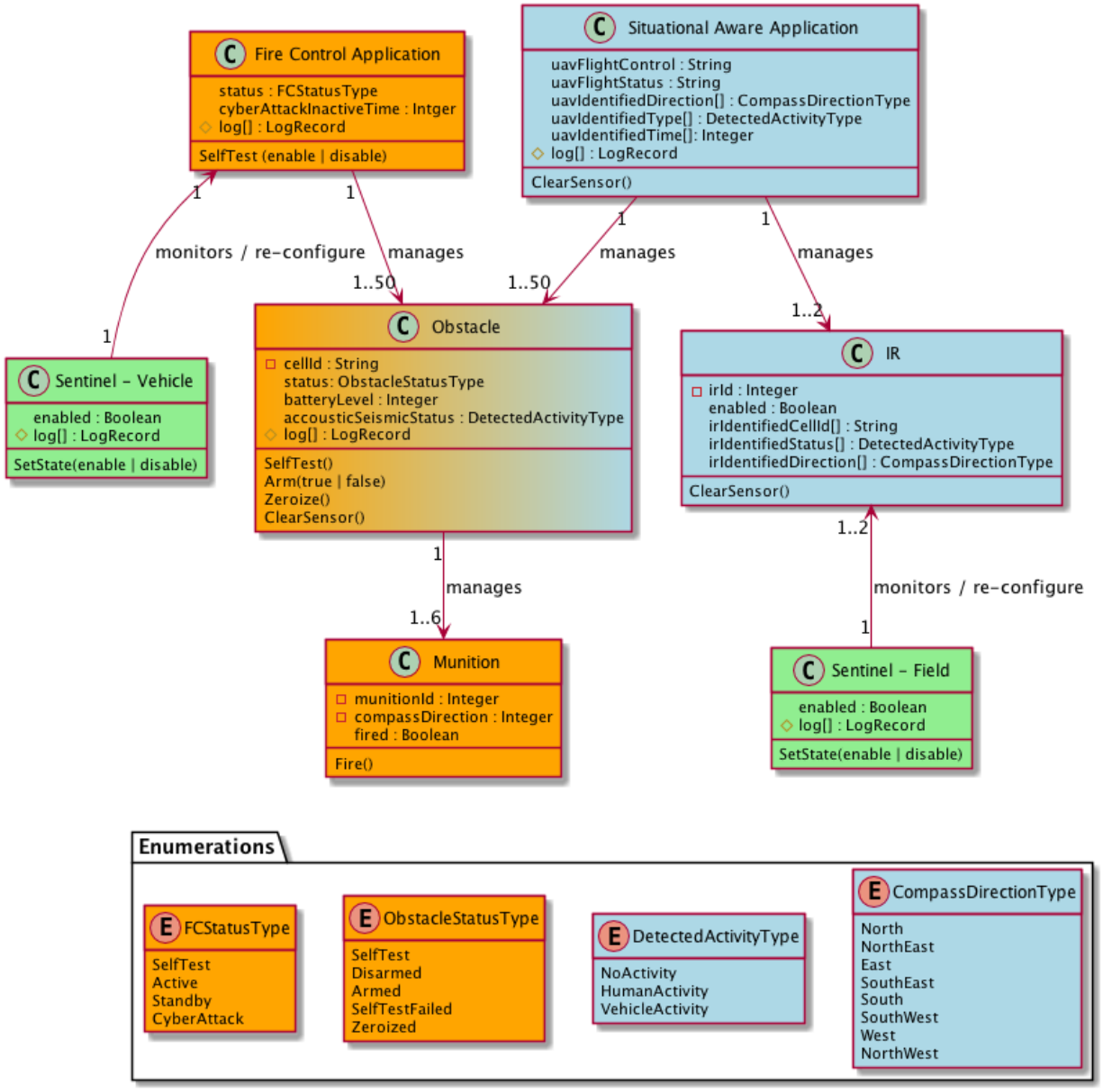

Figure 4.1.5 - Silverfish Data Model

Table 4.1.1.2 Silverfish Data Model

Object	Attribute / Operation / Relationship	Description
Fire Control Application		Fire Control Application (FCA) with diverse redundancy in an Active / Standby configuration
	(A) status	Active | Standby | SelfTest | CyberAttack
	(A) cyberAttackInactiveTime	FCA system inactive timer – from cyber-attack detected until operator enables resilient configuration
	(A) log[]	Log of Fire Control operations used for post mission forensic analysis
	(O) SelfTest ()	Initiate Fire Control Application self-test
	(R) manages	FCA manages up to 50 Obstacles (Fire Control Subsystem)
Obstacle		Ruggedized, tamer-proof hardware component which houses Fire Control munitions and Situational Aware sensors.
	(A) cellId	The geographic cell location (A1…G7) of the Obstacle
	(A) status	Armed | Disarmed | SelfTest | SelfTestFailed | Zeroized
	(A) batteryLevel	0 to 100%
	(A) accousticSeismicStatus	NoActivity | HumanActivity | VehicleActivity
	(A) log[]	Log of Obstacle operations used for post mission forensic analysis.
	(O) SelfTest()	Initiate Obstacle self-test.
	(O) Arm ()	Arm Obstacle (Enable Munition Firing)
	(O) Zeroize ()	Destroy (brick) the Obstacle to prevent recovery by adversary
	(O) ClearSensor ()	Reset current sensor values
	(R) manages	Obstacle manages up to 6 Munitions
Munition	(A) munitionId	Munition Id (1…6)
	(A) compassDirection	Fire compass direction – set at deployment time
	(A) fired	True | False
	(O) Fire()	Initiate munition fire
Situational Aware Application		Situational Aware Application (SAA)
	(A) uavFlightControl	Simulated Unmanned Aerial Vehicle (UAV) flight control settings
	(A) uavFligh Status	Simulated UAV flight status
	(A) uavIdentifiedDirection[]	Compass directions of UAV identified Physical Attackers [Array]
	(A) uavIdentifiedType[]	Type (Human | Vehicle) of UAV identified Physical Attackers [Array]
	(A) uavIdentifiedTime[]	Time distance of UAV identified Physical Attackers [Array]
	(A) log[]	Log of SAA operations used for post mission forensic analysis
	(O) ClearSensor()	Reset current sensor values
	(R) manages (IR)	SAA manages 1 or 2 Infrared (IR) sensors
	(R) manages (Obstacle)	SAA manages up to 50 Obstacles (Sensor Subsystem)
IR		IR Sensor with 360 degree view of entire protected field with diverse redundancy in an Active / Active configuration.
	(A) irId	1…2
	(A) enabled	True | False
	(A) irIdentifiedCellId[]	Geographic cell location (A1..G7) of IR identified Physical Attackers [Array]
	(A) irIdentifiedStatus[]	Type (Human | Vehicle) of IR identified Physical Attackers [Array]
	(A) irIdentifedDirection[]	Approach direction of IR identified Physical Attackers [Array]
	(O) ClearSensor()	Reset current sensor values
Sentinel – Vehicle		Sentinel deployed within the Vehicle wired network
	(A) enabled	True | False
	(A) log[]	Log of Sentinel operations used for post mission forensic analysis
	(O) SetState()	Enable or Disable the Sentinel – allows demonstration of cyber-attacks with and without the Sentinel

Object	Attribute / Operation / Relationship	Description
	(R) monitors / re-configures	Vehicle Sentinel monitors / reconfigures the FCA (as determined by the Blue Team priorities)
Sentinel - Field		Sentinel deployed within the Field wireless networks
	(A) enabled	True | False
	(A) log[]	Log of Sentinel operations used for post mission forensic analysis
	(O) SetState()	Enable or Disable the Sentinel – allows demonstration of cyber-attacks with and without the Sentinel
	(R) monitors / re-configures	Field Sentinel monitors / reconfigures the IR Sensors (as determined by the Blue Team priorities)

4.1.2 Silverfish Prototype Architecture Overview

4.1.2.1 Hardware Architecture

The Silverfish Prototype is built using a distributed network of Raspberry Pi computers as shown is the diagram below. A picture of the iTAR lab setup is shown on the following page.

The vehicle components are connected via a wired IP subnet. The field components are connected via redundant wireless subnets implemented using USB Wi-Fi dongles.

As shown in the iTAR lab picture, LED's are used to visualize redundancy state, munition state, and self-test status.

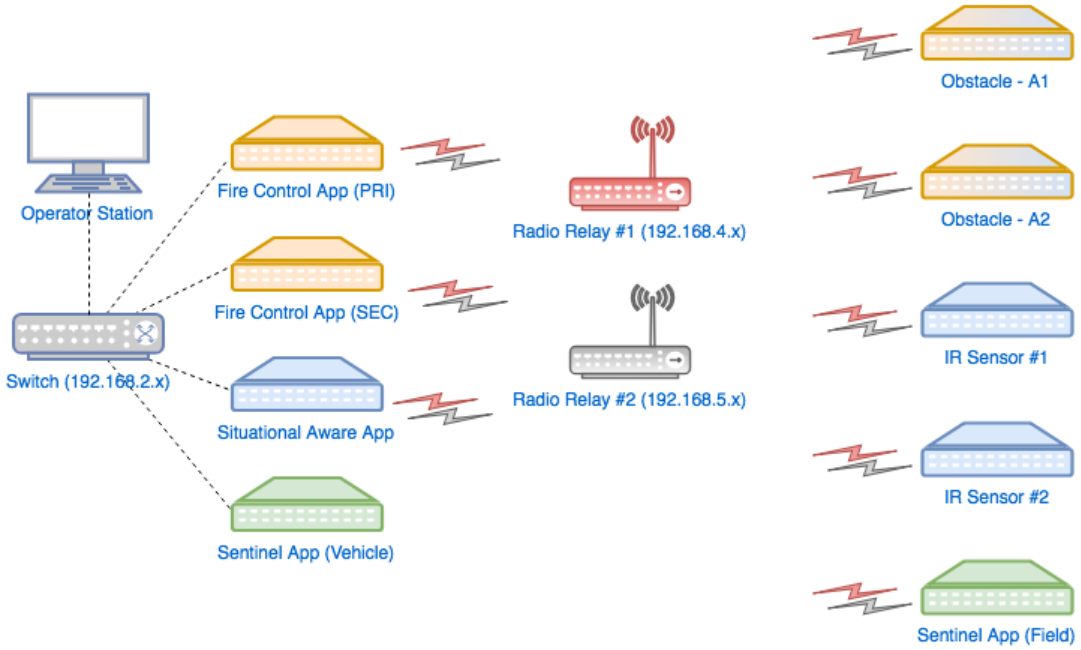

Figure 4.1.6 - Silverfish Hardware Components

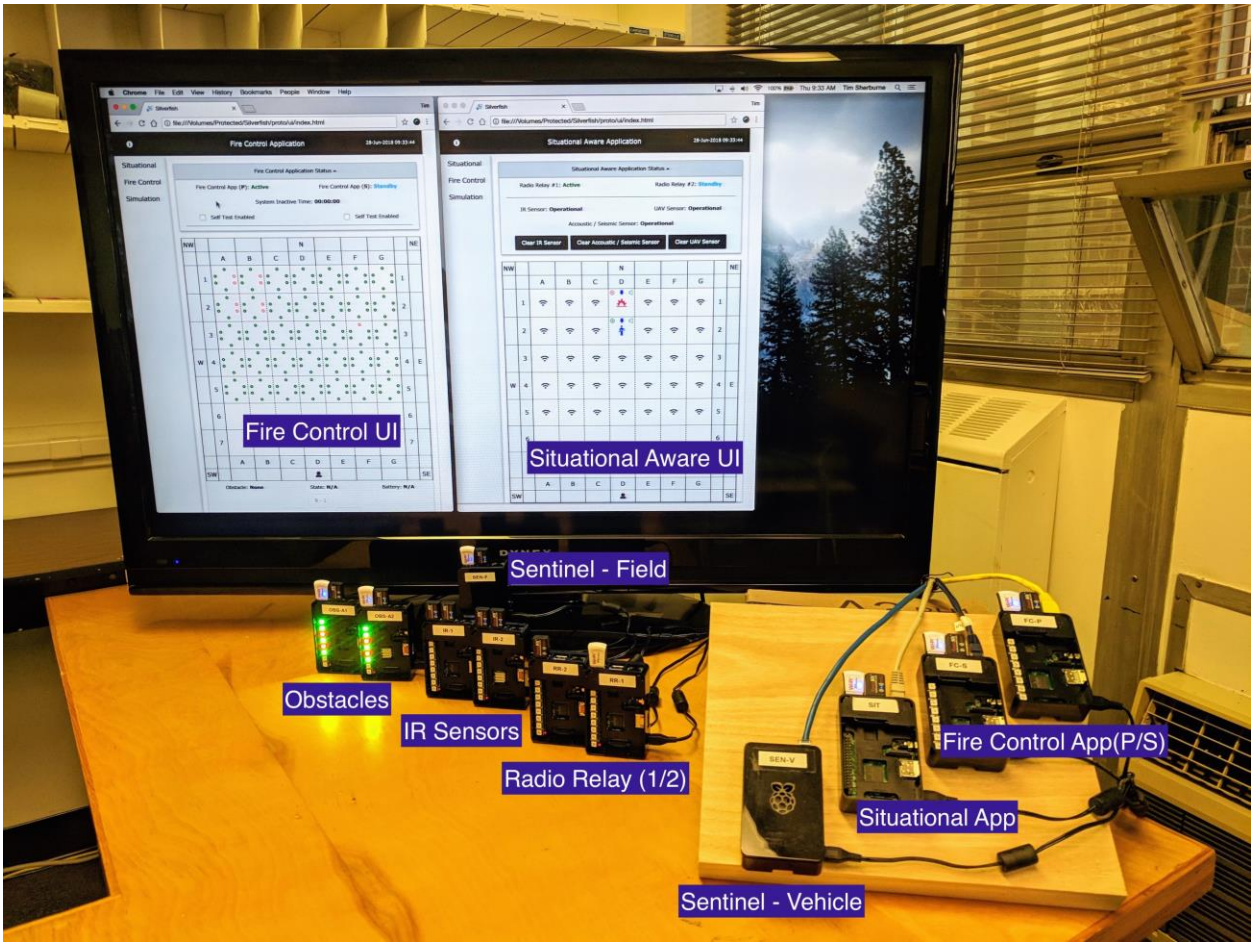

Figure 7.1.4 – Silverfish iTAR Lab Setup

4.1.2.2 SILVERFISH SOFTWARE ARCHITECTURE

As shown below, each Raspberry Pi runs the latest Raspbian Linux distribution and all back-end components are implemented using Python. The User-Interfaces are single page web applications implemented using HTML, CSS and JavaScript. All inter-process communication is implemented with a publish / subscribe messaging pattern using the Eclipse Mosquitto MQTT

message broker. The Eclipse Paho project provides the Python and JavaScript (via WebSockets) clients.

The publish / subscribe messaging pattern enables both request / response messaging and asynchronous notifications by simple topic tree design. The flexibility and simplicity of the topic tree is well suited for rapid prototyping.

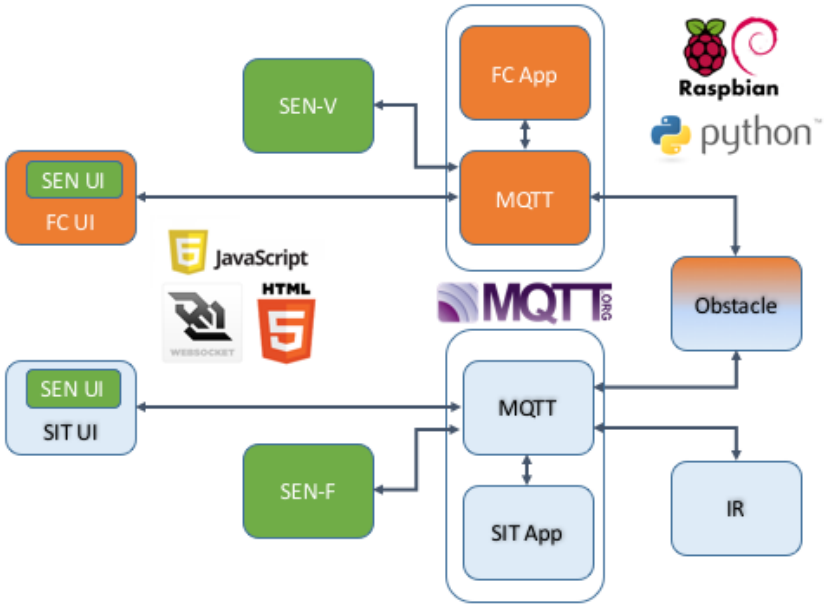

Figure 4.1.8 - Silverfish Software Architecture

4.1.2.3 SILVERFISH USER INTERFACE OVERVIEW

To simplify the User Interface prototype a single User Interface is developed with a left-hand navigation to launch the Silverfish Application views (Situational Aware, Fire Control, & Simulation Control). The following pages show each view with a brief explanation of key functionality. Each application view shares the grid-view display concept with varied grid content. The grid supports individual cell selection as well as a "lasso" multi-select.

The Fire Control Application User Interface is shown below. Each grid cell (A1-G7) displays the state of the Obstacle Munitions (green – ready to fire, red-fired). The information panel below the grid provides controls for the selected cell(s) including the ability to fire one or more

munitions, arm / disarm the obstacle, initiate a set of obstacle self-tests, and the ability to zeroize an obstacle.

The drop-down panel at the top of the grid displays the redundancy status (active / standby) of fire control application with controls for switching.

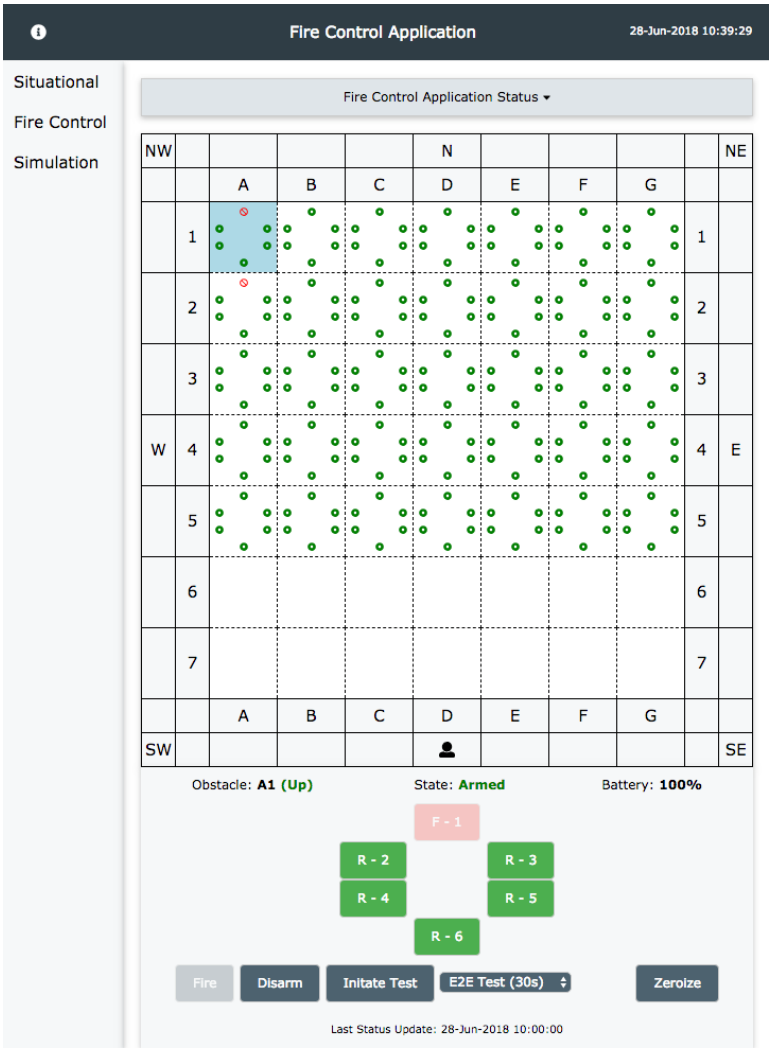

Figure 4.1.9 - Fire Control User Interface

The Situational Aware Application User Interface is shown below. Each grid cell displays the sensor reporting status. For example, cell D1 shows vehicle activity, moving in from the north, confirmed by both IR and Seismic / Acoustic sensors.

The boundary of the grid shows the compass directions while the icon at the bottom indicates that the operator is observing the protected field from the south looking north. The UAV provides early warning activity reports with icons on the grid boundary showing type, direction and estimated time distance of a physical attacker.

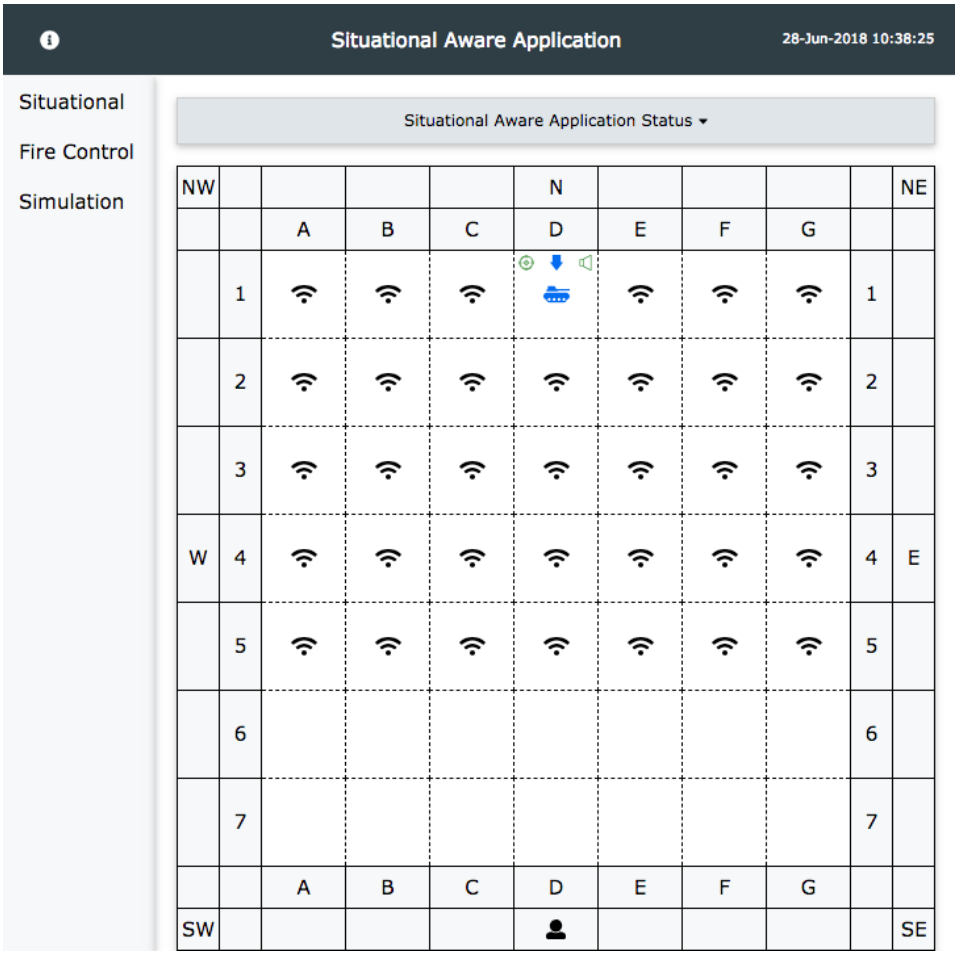

Figure 4.1.10 - Situational Aware User Interface

The Simulation Control User Interface is shown below. Each Grid cell represents a deployed Raspberry Pi obstacle. The information panel above the grid provides controls for the selected cell(s) including resetting (after firing / zeroizing), changing battery level, controlling self-test pass / fail, as well as acoustic-seismic and IR sensor activity type.

The drop-down panels at the top of the grid provide controls for initiating cyber-attacks and controlling UAV sensor reporting.

To provide an end-to-end sensor scenario of physical attackers approaching, entering and moving though the protected area, a CLI-based scripting interface is also provided.

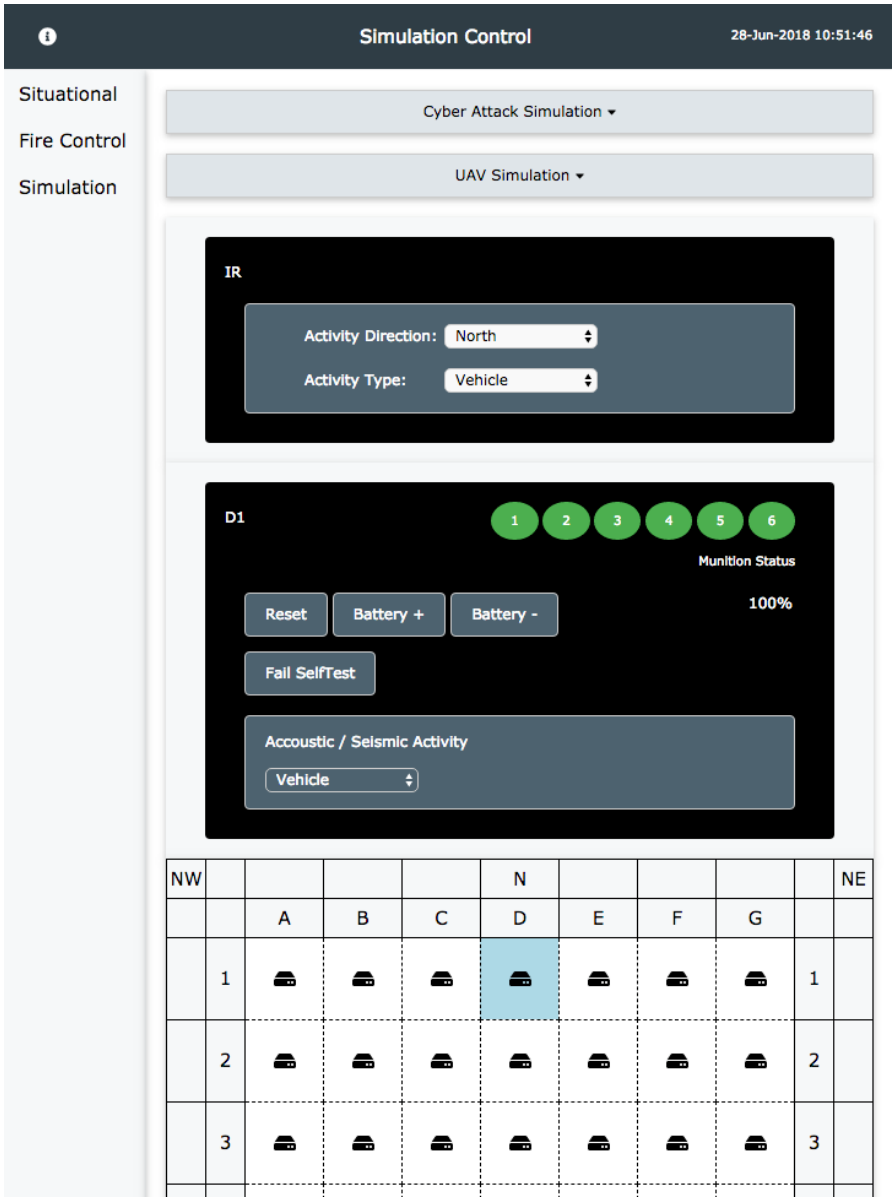

Figure 4.1.11 - Simulation Control User Interface

4.1.3 SILVERFISH CYBER ATTACK / SYSTEM RESILIENCY USE CASES

The following Cyber Attack and System Resiliency uses cases where driven by the Blue Team prioritized list of "system functional problems to be avoided". Additionally, the use cases where

defined such that a varied set of Cyber Attack design patterns and system resiliency solutions were demonstrated.

4.1.3.1 USE CASE SUMMARY

Table 4.2.3.1 Use Case Summary

Title	Description	Detection Method / Corrective Action
1.1 Inappropriate Firing via Manipulated Operator Commands **Attack Target**: Fire Control Application **Attack Method**: Insider – SW Developer	During design and manufacture, a SW Developer introduces software to the Fire Control Application that redirects Operator fire commands when deployed at a specific geographic location. With this Cyber-attack knowledge, a Physical attacker could gain access to a protected area. The Fire Control Application includes Primary and Secondary instances which are based on independent design and manufacture so as to minimize the likelihood of the same Cyber Attack affecting both.	*Design Pattern: Changing Control Input* Detection Method The Sentinel Application within the Vehicle monitors the Fire Control Application for consistency between Operator requested actions and the actions that will be delivered to the Obstacles via the Radio Relay Interface. Corrective Action The Sentinel detects the attack and takes the following actions: • The misfire is aborted. • The Primary Fire Control Application is taken out of service and put into a "CyberAttack" state. • The Secondary Fire Control Application is put into a "SelfTest" state. To gain confidence with the reconfigured system, the Operator takes the following actions: • Individually test one or munitions. • Multi-Select a group of munitions for test. • If and when confidence is restored, Activate the Resiliency Mode (disable the "Self Test" of the Secondary Fire Control Application) and continue operation.
2.2 Prevent or corrupt transmission of situational awareness data **Attack Target**: Radio Relay **Attack Method**: External	During operation of the Silverfish network, a Cyber Attacker gains access to the Radio Relay network and injects false IR sensor report messages. The Silverfish network includes Primary and Secondary Radio Relay instances which are based on independent design and manufacture so as to minimize the likelihood of the same Cyber Attack affecting both.	*Design Pattern: Introspection* Detection Method The Sentinel Application within the Field monitors network traffic and maintains a profile of "normal" traffic loads based on current field state. Corrective Action The Sentinel detects a higher than normal level of IR sensor reporting activity. The Sentinel disables the Primary Radio Relay network changing its state to "Disabled" thereby notifying the Operator of the potential Cyber Attack. The Sentinel attempts to isolate the Cyber Attack by activating the Secondary Radio Relay network while continuing to monitor the IR sensor reporting activity

Title	Description	Detection Method / Corrective Action
		level. If the level returns to normal, the Sentinel marks the Primary Radio Relay network as ”CyberAttack” thereby notifying the Operator of the confirmed attack and Corrective Action taken.
2.1 Delays in situational awareness **Attack Target**: IR Sensor **Attack Method**: Insider	During design and manufacture, a SW Developer introduces software to the IR Sensor that delays sensor reports when deployed at a specific geographic location. With this Cyber-attack knowledge, a Physical attacker could gain access to a protected area. The IR Sensor subsystem includes two instances which operate in an Active / Active configuration and are based on independent design and manufacture so as to minimize the likelihood of the same Cyber Attack affecting both. Each Sensor is capable of monitoring the complete field.	*Design Pattern: Data Consistency* Detection Method The Sentinel Application within the Field monitors Sensor Activity for consistency (Seismic-Acoustic, IR1 / IR2 & UAV). Corrective Action The Sentinel Application detects ongoing inconsistencies between IR1 and IR2 / Seismic-Acoustic Sensors. The Sentinel “votes” IR1 sensor as tampered, disables it. and sets it state to “CyberAttack” thereby notifying the Operator of the Cyber Attack. The Situational Aware Application continues to operate in a “reduced” state based on the single IR sensor reports. The Situational Aware application recommends that an additional Corrective action would be for the Operator to relocate the vehicle and / or the UAV to a better vantage point for manual observation.

4.1.3.2 REALIZATION OF USE CASE 1.1 – MANIPULATED OPERATOR COMMANDS

The following message sequence diagrams (part 1 / 2) show the realization of "Manipulated Operator Commands" use case by the Silverfish Prototype.

The Vehicle Sentinel provides the single function of detecting inconsistencies between operator fire control requests and obstacle received fire control requests. The Sentinel automatically contains the cyber-attack and then instructs the operator on options for system reconfiguration.

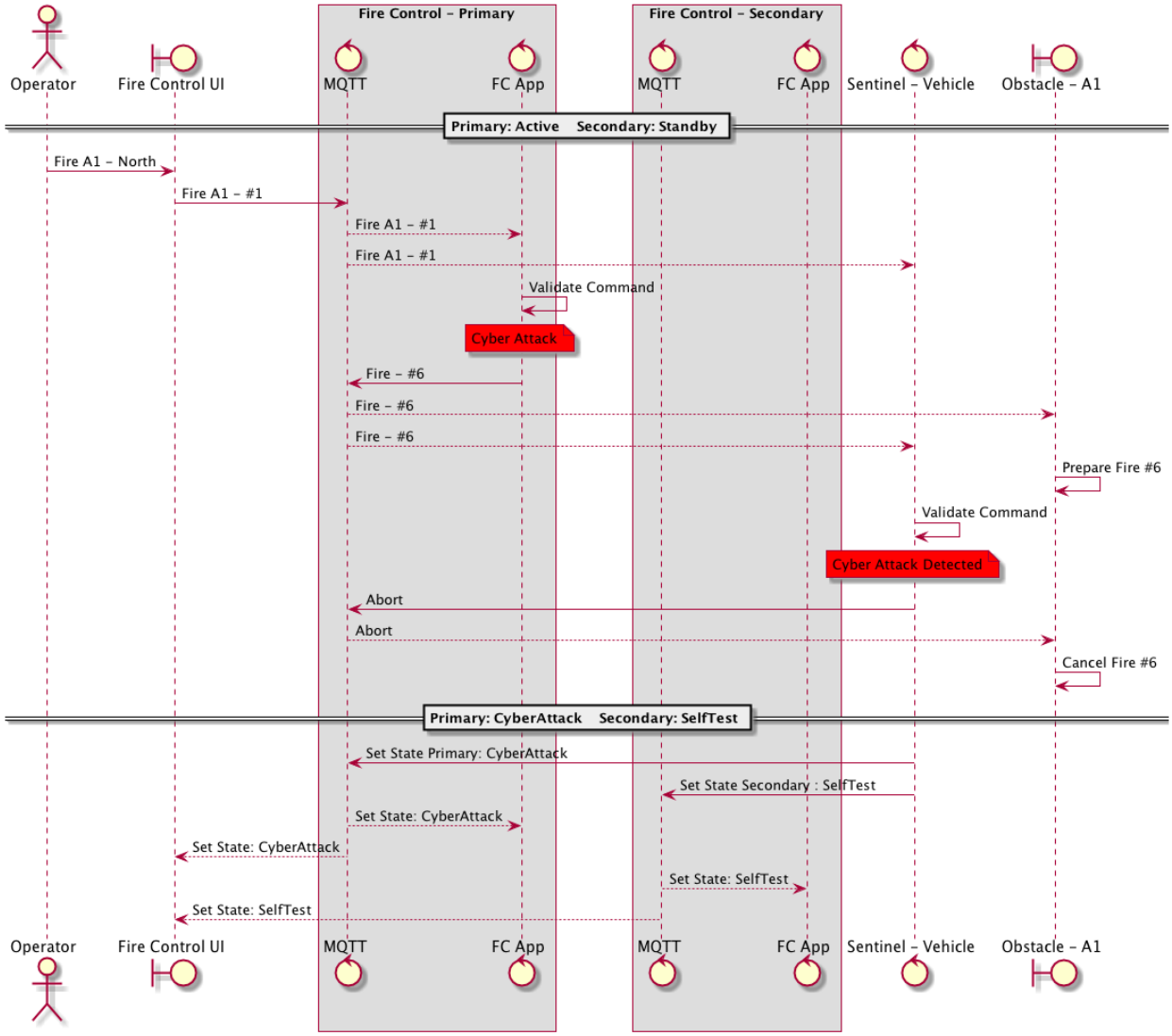

Figure 4.1.12 - Manipulated Operator Commands - Part 1

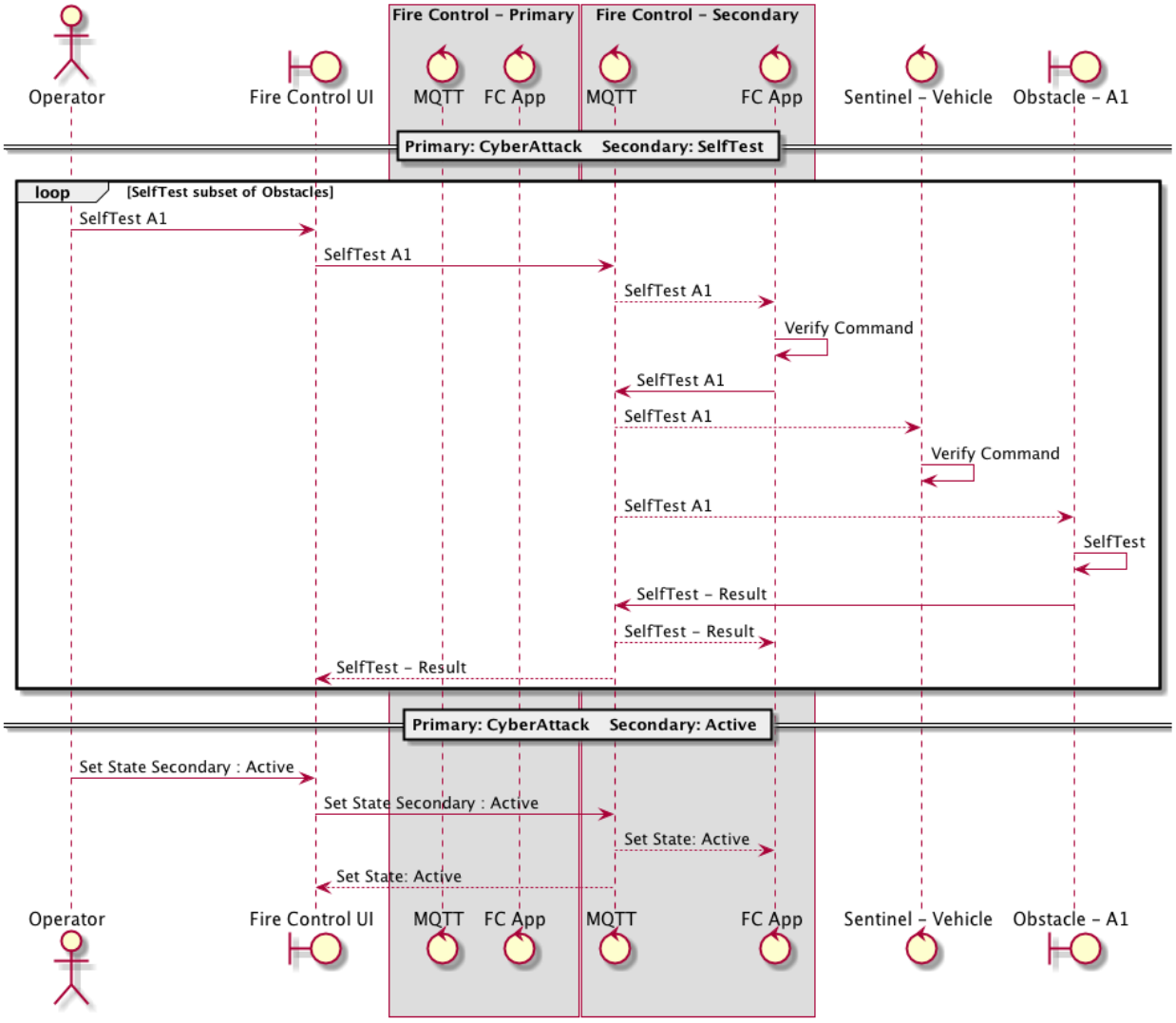

Figure 4.1.13 - Manipulated Operator Commands - Part 2

4.1.4 Specific Application of Rapid Prototyping Tools

As CSRM results materialized, the SE Team was stimulated to perform specific assessments. Of particular interest were the issues that resulted from cyber security-related recommendations to separate the hardware and software for the Silverfish situation awareness and weapon control functions. This recommendation results in the Silverfish operator needing to utilize two separate displays to execute his or her roles. Rapid prototyping served to demonstrate human factors-related issues that could emerge from this potential requirement. In addition, the newly developed resilience design pattern that permits operators to conduct tests to confirm the reconstitution of the attacked system also raises issues regarding which display to utilize for presentation of test results. Use of the situation awareness display would better assure the security of the control portion of the Silverfish system whereas use of the weapon control display

would keep the operators' attention on test cases involving weapon control assurance. Both of these situations would require decisions that related to trade-offs between user performance and cyber security.

4.2 Analysis Tools

The CSRM makes use of a small set of support tools to facilitate interactions between team members and to complete the tasks associated with each step of the process. These tools include No Magic's MagicDraw software for creating and modifying system models in the Systems Modeling Language (SysML), a Systems Theoretic Accident Model and Process (STAMP) hazard analysis methodology to help identify key requirements, critical functions, and organize modeling efforts, and the Cyber Body of Knowledge (CYBOK) tool developed by VCU to support the identification and quantification of the likelihood of attacks for the Red Team.

4.2.1 SysML and MagicDraw

The Systems Modeling Language is a general-purpose, graphical modeling language standardized by the Object Management Group (OMG) for model-based systems engineering. Based on UML, SysML is designed to be more abstract and flexible, which allows for its application to systems beyond just their software. SysML is based on a set of diagram types with an associated set of diagram elements that follow a specific syntax with clear semantics. These diagrams represent the structural composition and interconnections between architectural structures, admissible behaviors, requirements, and the relationships between these elements within a system.

The CSRM uses SysML to document the design of the Silverfish system and support the tasks of the Blue, Red, and SE teams. An initial, simple SysML description of the system is created in concert with activities of the Blue Team during the Step 1 of the CSRM. This initial description is then augmented and adjusted throughout the entire CSRM process as design decisions are explored and evaluated. SysML is a powerful tool for facilitating the communication and understanding of design elements within a system between different stakeholders. Graphical representations of architectures, behaviors, and requirements combined with the ability to define relationships between elements makes it much easier to both describe and understand the effect of specific design choices on a system.

The MagicDraw software used to develop the SysML model of Silverfish enables quick model construction and adjustment. This allowed the model to be updated in near real-time as design choices were agreed upon by the stakeholders. Additionally, SysML software such as MagicDraw, enables modelers to encode much more detail into system models "behind-the-scenes" than what is visible in the main graphical representations. For example, model elements in different diagrams can be linked to one another via a trace relationship when defining the model element in the software tool. This allows the modelers to keep track of and communicate the interactions between various elements across the system more easily. An example of one such trace is present in Figure 4.2.1.

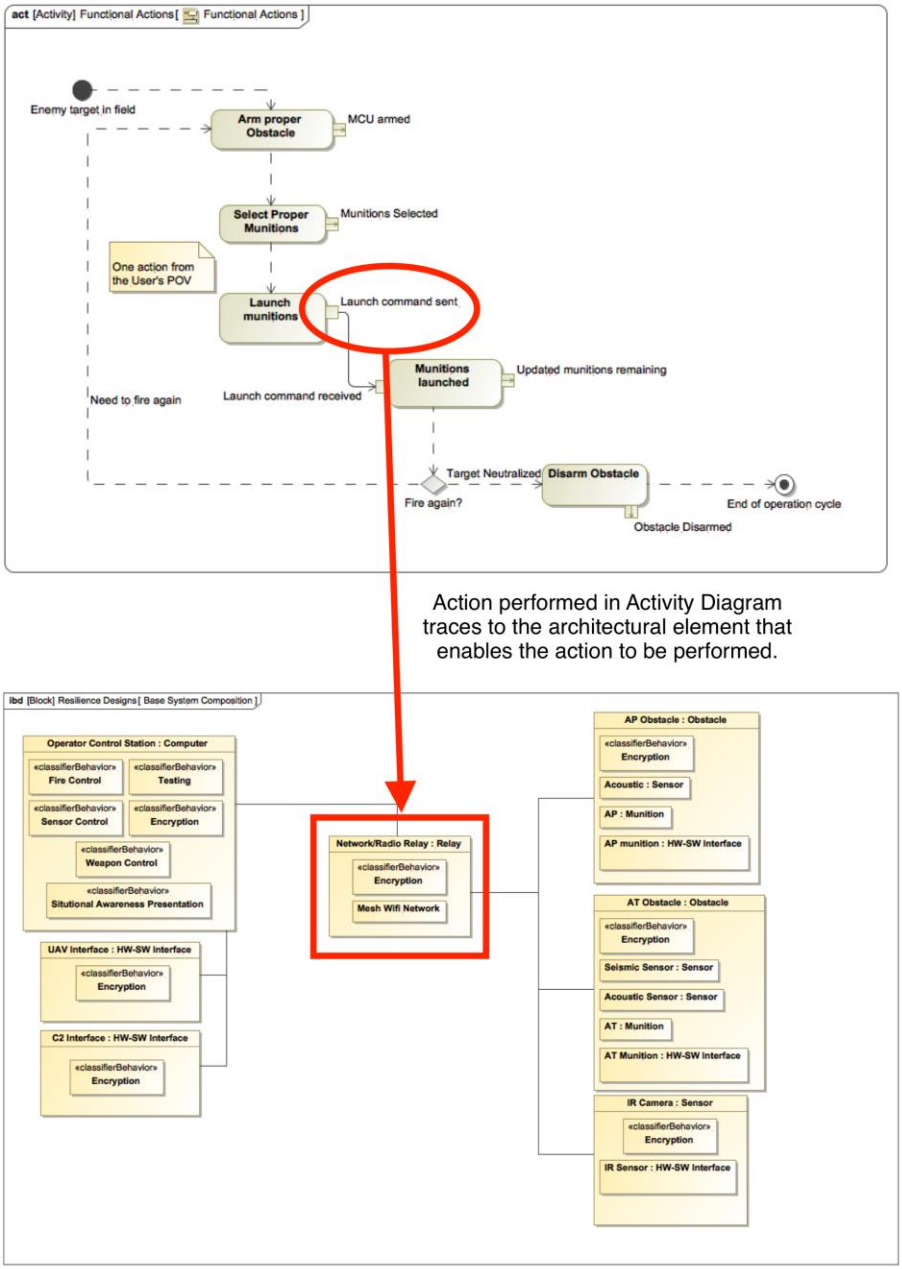

Figure 4.2.14 - An example of a cross-diagram trace within the SysML model of Silverfish

4.2.2 STAMP-BASED HAZARD ANALYSIS

Systems Theoretic Accident Model and Process (STAMP) is an accident causality model developed by MIT that captures accident causal factors including organizational structures, human error, design and requirements flaws, and hazardous interactions among non-failed components. In STAMP, system safety is reformulated as a system control problem rather than a component reliability problem; i.e., accidents occur when component failures, external disturbances, and/or potentially unsafe interactions among system components are not handled adequately or controlled. STAMP further is founded on the assertion that system safety controls are hierarchical in nature, as is commands or control doctrine, with commands issued from higher levels of the military organization to lower levels, with feedback provided from lower levels to higher levels.

STAMP based hazard analysis tools, such as Systems Theoretic Process Analysis (STPA) and Systems Theoretic Process Analysis for Security (STPA-Sec), use the underlying assumptions of the STAMP model to postulate about the possible ways for accidents to occur in a system and facilitate the development of requirements and design choices that make those accidents less likely to occur. Both STPA and STPA-Sec methodologies begin with the identification of high-level losses that system owners would like to avoid, which is well aligned with the Blue Team consequence elicitation process conducted in Step 2 of the CSRM.

As such, the SE team uses the consequences identified by the Blue Team to initiate a version of STAMP-based hazard analysis that helps create requirements and define model structure within the SysML model of the system. More specifically, the Blue Team consequences are translated into requirements language along with some other requirements identified by STAMP-based methods to be documented and organized appropriately in the SysML model. Furthermore, the STAMP-based analysis helps identify a modeling structure for representing system behaviors and architectures that follows hierarchical pattern. Examples of the appearance of STAMP-based information in the SysML model are shown in the SysML diagrams presented in section 5 of this report. More information about STAMP-based hazard analysis can be found in the SERC RT-172 Technical Report TR-114.

4.2.3 CYBOK

The Cyber Body of Knowledge is a VCU-developed tool that makes use of the MITRE attack databases, the Common Attack Pattern Enumeration and Classification (CAPEC), the Common Weakness Enumeration (CWE), and the Common Vulnerability Enumeration (CVE). In general, CYBOK compares the contents of a system model against entries in these databases to help develop an understanding of the cyber-threats to the system. In the CSRM process, CYBOK supports the Red Team efforts by bringing historical evidence of attacks against similar systems to the one being developed to further enhance the credibility of the experience-based Red Team activities. More information about CYBOK can also be found in the SERC RT-172 Technical Report TR-114.

5. APPLICATION OF CYBER SECURITY REQUIREMENTS METHODOLOGY TO SILVERFISH USE CASE

This Section will sequentially present the results for each of the CSRM steps for the Silverfish Application presented in Section 3. As described earlier in this paper, the SE team orchestrated the overall process, but different teams managed the activities associated with the individual steps.

5.1 CSRM STEP 1

This step, managed by the SE team (UVA), resulted in the SysMl description of an initial Silverfish architecture. This step was explicitly defined not to include a plan to account for cyber security considerations, leaving that effort to a later stage in the CSRM process. The activity to create the system description involved regular interactions between the SE and Blue Teams over a 3-month period. The SE team addressed the Silverfish system's technical design, depending on the Blue Team to support their understanding of operational considerations. An important outcome of this interaction was the unanticipated desire of the Blue Team to have Silverfish communications encrypted. This desire was based upon experience with the operations of legacy systems related to the Silverfish mission, and the strength of their commitment to assure that an adversary could not either take control or corrupt control of weapon firings. While the CSRM concept did not look to the Blue Team for cyber security inputs as part of Step 1, it was agreed that encryption would be included in the initial Silverfish system description and would be later scrutinized by the Red Team as part of Step 4.

Step 1 of the CSRM resulted in the creation of the initial SysML representations of the baseline Silverfish system as described in section 3.2 of this report. At this early stage of development, the SysML model largely serves to document and communicate the main functionality and basic architecture of the Silverfish system. Discussions with the Blue Team over the course of the 3-month period in Step 1 led to incremental additions to the SysML model as the baseline architecture to the Silverfish system was defined and agreed upon. Important additions to the system in Step 1 include the introduction of encryption across the main components of Silverfish and the specification of the particular sensors to be included for the surveillance function of Silverfish. These two additions can be seen in the internal block diagram of the system composition in Figure 5.1.

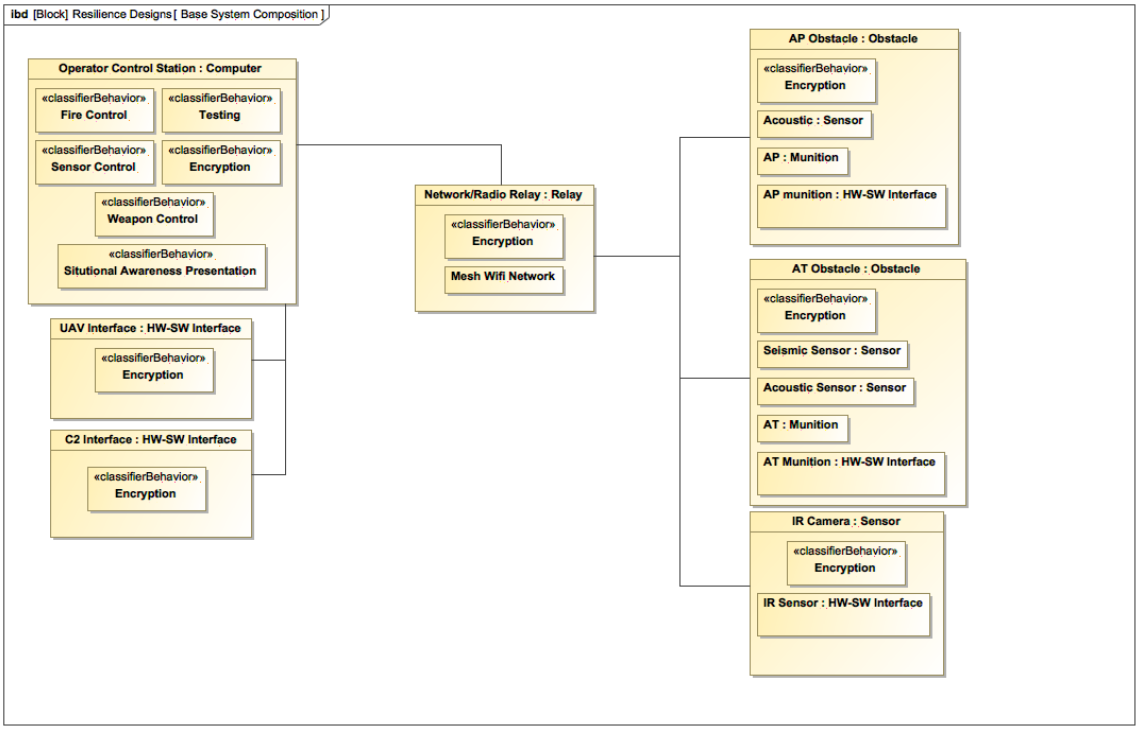

Figure 5.15 - An internal block diagram of the baseline Silverfish architecture. Note the inclusion of encryption across all major components.

Moving forward in the CSRM process, these initial SysML descriptions are augmented as design decisions are reached and security concerns addressed. The model serves as a consistent point of reference for the SE team and Blue Team. The basic model of the system at this stage creates the basis for the STAMP-based hazard analysis to be performed by the SE team.

As part of Step 1, the SE team initiated development of the rapid prototype/simulation vehicle described in Section 4.1. This effort will serve to enable evaluation of the system's performance, with emphasis on the operator's interaction with the Silverfish situation awareness and weapon control display subsystem.

5.2 CSRM STEP 2

This step, managed by the Blue Team (ARDEC), resulted in identifying and prioritizing a set of unintended operational use consequences that they would want to be avoided in the application of Silverfish (e.g., operator weapon control commands manipulated by an adversary). The assessment was based on an analysis of the SysMl Silverfish description provided by the SE team. The process used to create the prioritized consequence table involved a 5-hour interaction between the SE team and Blue Team, with the SE team providing clarifying information regarding the Silverfish technical design choices and the Blue Team identifying and prioritizing the Silverfish use consequences to be avoided. It is noted that the earlier CSRM Step 1 interactions between

the SE and Blue Teams regarding the Silverfish system design significantly reduced the time needed to conduct the use consequence prioritization effort.

Table 5.1 below presents the set of 11 use consequences identified and prioritized by the Blue Team. A Likert scale was used to cluster the use consequences into 4 distinct groups regarding seriousness. While there were additional Likert Scale 5 use consequences identified, the completeness of the Silverfish design was insufficient to address them and the scope of the research effort was too bounded to broaden the Silverfish design effort. This limitation was not considered to be of sufficient importance so as to distort the conclusions regarding the CSRM. In addition, as described in Section 3, Table 4.2 identifies the STAMP types that SE Team associated with each of the consequences described by the Blue Team.

Table 5.3 Results of Blue Team Consequence Prioritization Process

ID	Attack Outcome	Attack Target(s)	Attack Method	**STPA Type**	Likert Priority
1.1	Inappropriate firings via manipulating operator commands	Operator control display, radio comm links	External, supply chain, insider	1, 2, 3	1
1.2	Delays in fire time (sufficient delay to cross field) – Includes Denial of Service	Obstacles, control station, radio comm links	External, supply chain, insider	2, 3	1
1.3	Delays in deployment	Obstacles, deployment support equipment	Supply chain, insider	2, 3	1
1.4	Deactivation of a set of obstacles	Obstacles	External, insider	1, 3	1
2.1	Delays in situational awareness	Operator display, sensors, Radio comm links	External, insider, supply chain	1, 2, 3	2
2.2	Prevent or corrupt transmission of situational awareness data	Radio comm links, operator display, sensors	External, insider, supply chain	1, 2, 3	2
2.3	Gain information to help adversary navigate through field	Obstacle, operator control station	External, insider	2, 3	2

3.1	Reduced operational lifespan –battery rundown	Obstacle	External, supply chain, insider	1, 2, 3,	3
3.2	Prevent transmission/execution of non-firing commands	Operator display, obstacles	External, insider, supply chain	1, 2	3
4.1	Delays in sending/receiving C2 information	Operator display, radio comm links	External, supply chain	1, 2, 3	4
4.2	Delays in un-deployment	Obstacles	External, insider, supply chain	1, 2, 3	4

STPA hazard types

1. *Providing a control action causes a hazard*
2. *Not providing a control action causes a hazard*
3. *Incorrect timing or improper order of control actions causes a hazard*
4. *A control action is applied too long or stopped too soon*

Likert Priority Scale

1. *Unacceptable and highest priority to provide resiliency*
2. *Avoid as long as resiliency solution does not over-complicate operation*
3. *Would like to avoid, but solution needs to be incremental*
4. *Lowest priority, low-cost, simplistic solutions should be considered*
5. *Not of interest at the present time, recorded for future use*

Following the Blue Team exercise, the SysML model is augmented to include the consequence prioritization in a requirements diagram. The consequences are translated into requirements language by the SE team and organized in a hierarchical fashion that mirrors the rankings from the Blue Team. Additional requirements are derived by the SE team using STAMP-derived formulations based on the initial descriptions of the Silverfish system's purpose and functionality. The requirements diagram can be seen in Figure 5.2.

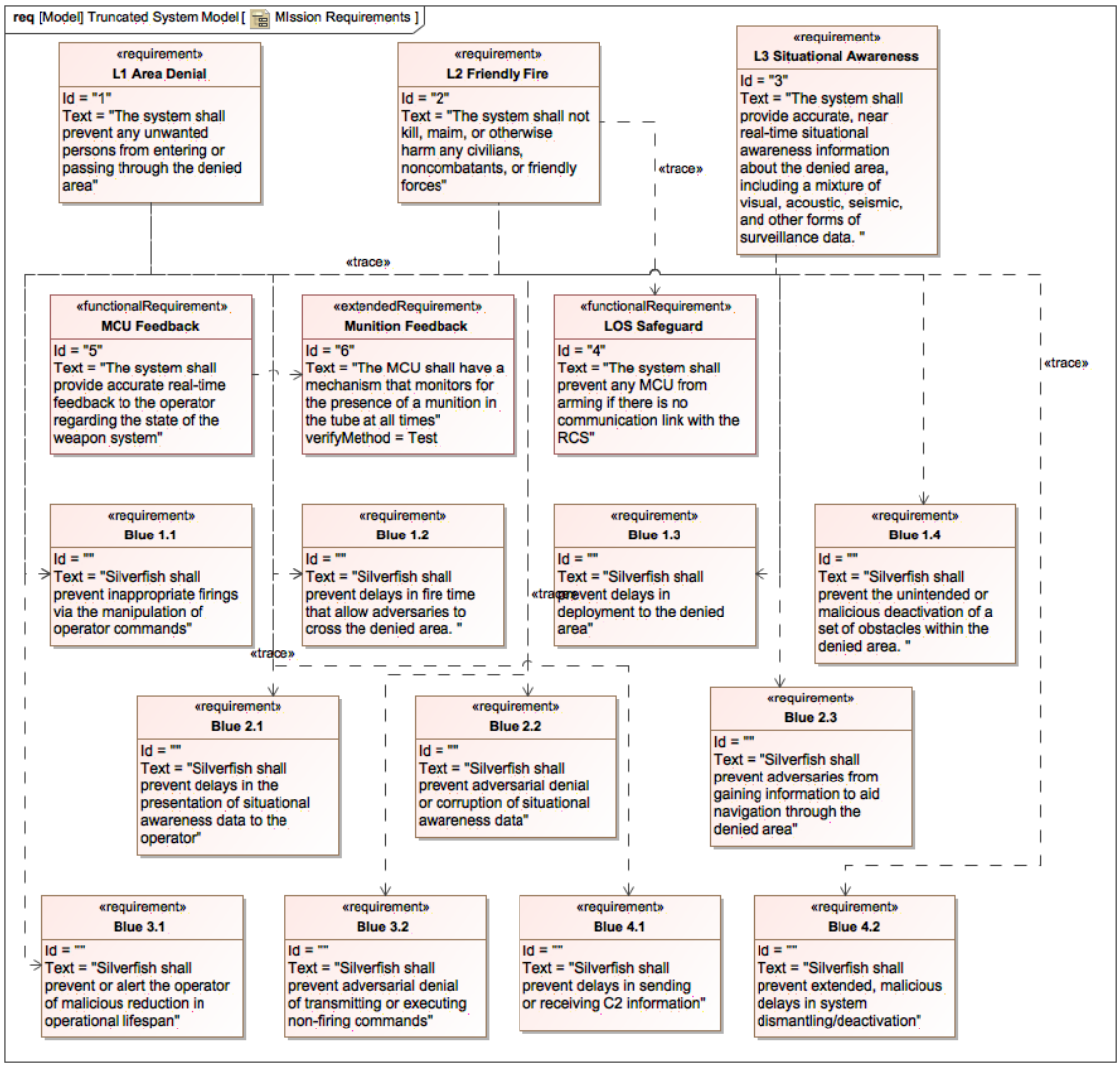

Figure 5.16 - Requirements diagram based on the output of the Blue Team exercise.

In Figure 5.2, the Blue Team consequences are given names such as “Blue 2.1” that correspond to their ranking in the exercise. The additional STAMP-derived requirements are located near the top of the diagram and serve as root nodes from which the other requirements can be traced.

The following Section describes how this tabular result is utilized by the SE Team in Step 3 of CSRM.

5.3 CSRM STEP 3

Step 3 of the CSRM involves the SE Team developing potential resilience solutions that respond to the Blue Team results presented in Table 5.1 above. For the purposes of this research activity, it was decided to develop three (3) different options for providing resilience, recognizing that any

combination of these options could ultimately be decided upon as a cyber security resilience requirement. The three areas for resilience selected by the SE Team were:

- Resilient weapon control capabilities (including data consistency checking design pattern, and diverse redundant HW/SW) implementation for the operator's vehicle-mounted computer)
- Diverse redundant communications sub-systems,
- Resilient situation awareness capabilities (including diverse redundant sensor voting and situation awareness introspection design patterns)

An important aspect in selecting potential resilience requirements were the results that emerged from the rapid prototyping/simulation efforts of the SE Team, initiated as part of Step 1 of CSRM. For all three cases presented above, it was decided to include the newly developed design pattern referred to as "Real-Time Resilience Confidence Testing". This design pattern provides the operator with the opportunity to initiate pre-designed tests of the diverse mode of operation intended to be used as a basis for achieving resilience. The tests can range from a trial of the new component about to be put into use, to an end-to-end trial that includes the new component operating together with the other system technical components that would be involved in performing the system function that is about to be restored. A successful test would serve to increase operator confidence that the restoration will indeed provide the desired resilience, but it comes at the cost of using up time that could be operationally critical depending upon the attacker's intent. Ultimately, the operator must decide on the trade-off between confidence in achieving the desired resilience outcome and the use of time for testing, based upon the battlefield context of the mission that the protected system is engaged in.
In addition, the introspection design pattern referred to above as "Situation Awareness Introspection" was also newly developed as part of this research effort. This resilience feature involves making comparisons between the level of situation awareness activity being displayed to the operator and the level of associated machine utilizations that relate to those displays (e.g., CPU utilization, memory access activity, and communications network traffic levels related to detected adversarial activities). For example, if the operator's situation awareness display shows no adversarial information, but the network is receiving significant numbers of surveillance packets from the various sensor systems, there is strong reason to suspect a cyber attack.

Throughout the 6-step CSRM, the Silverfish Prototype and Simulation provided valuable insights. Highlights include:

Human Factors: Early system design decisions included the possibility of a security related design requirement to isolate the Fire Control and Situational Aware applications and their associated User Interfaces. However, during prototype demonstrations, potential operator usability concerns emerged. It was observed that the process of an operator switching his/her attention between two separated user interfaces while addressing concurrent cyber and physical attack from an adversary can potentially lead to confusion and inefficiencies. It was concluded that prior to decision-making regarding the separation of displays it would be important to explore this

design issue in more detail, building on the use of the rapid prototyping/simulation vehicle for human factors related evaluations.

Sentinel Interfaces and Timing: During the specification of cyber attack and system resiliency use cases, it is necessary to consider system interfaces and system timing requirements related to the Sentinel achieving its intended functions. An important example related to timing requirements arose during the conduct of a simulation scenario involving a cyber-attack that modified the operator's fire command. In order for the Sentinel to automatically abort corrupted commands the system timing requirements would have to allow time for both detection of such an attack as well as withholding the command until the Sentinel determines that it is an appropriate command. These timing allowances would serve to delay fire times; an undesirable system resiliency bi-product.

The SE Team developed SysML representations for each of the three Silverfish resilience options presented above. These would be the basis for proceeding on to Step 4 of the CSRM, involving the Red Team making assessments of the combined baseline Silverfish system architecture including potential resilience features for the system.

In Step 3 of the CSRM, the SysML models are augmented using additional internal block diagrams to outline the possible resilient design options for critique by the Red Team. Below, in Figure 5.3, additional functionality and components for resiliency in fire control are represented by the colored boxes added to the baseline Silverfish architecture. Specifically, the fire control resiliency option separates the operator control station in the baseline architecture into a situational awareness station and a weapon control station, represented by the green boxes. Additionally, the Sentinel (pink box) monitors the weapon control station and its redundant, diverse counterpart. Within the Sentinel box, the monitoring functions that it provides are defined by <<classifierBehavior>> blocks in the SysML taxonomy.

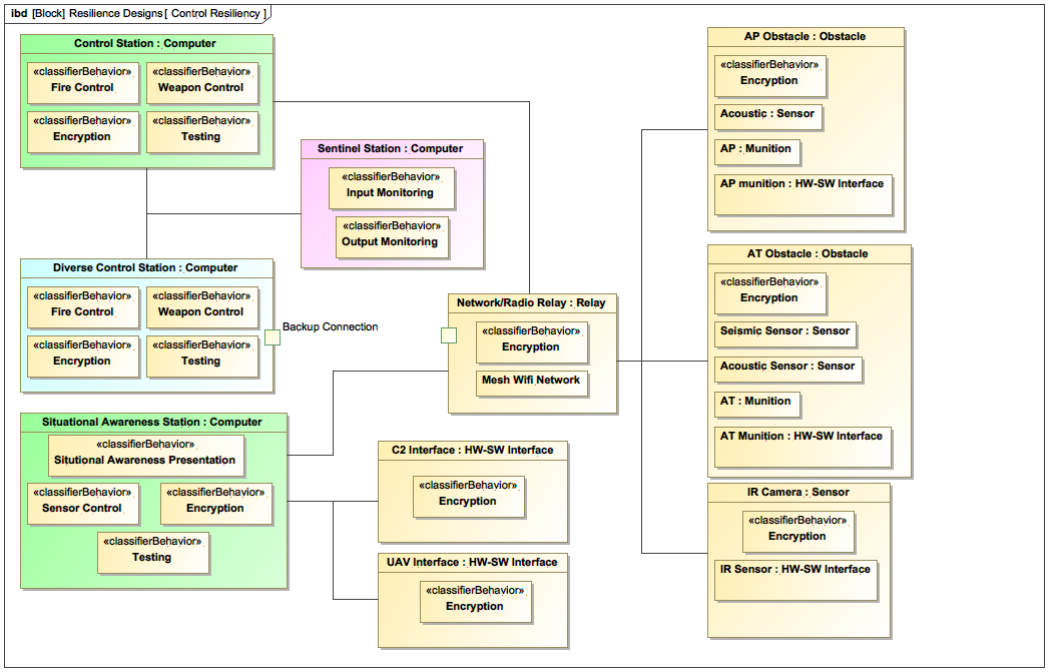

Figure 5.17 - Fire control resiliency architecture.

The second resiliency option, diverse communication, is presented in Figure 5.4. This option involves the inclusion of a redundant network for communication between the operator station and the obstacle field and an associated set of Sentinels that monitor for message content, rate, and metadata consistency across the system. Like in Figure 5.3, diverse, redundant components are represented by light blue boxes and Sentinel agents are represented by light pink boxes.

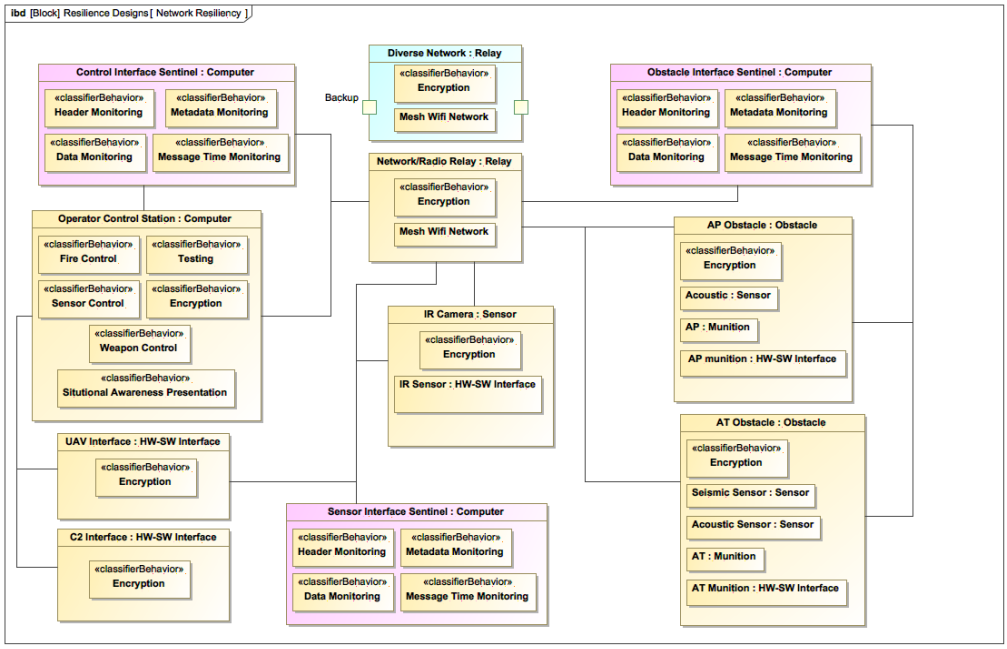

Figure 5.18 - Network resiliency architecture.

The final resiliency option explored, resilient situational awareness capability, is defined in Figure 5.5. This option mirrors the resilient weapon control option by splitting the operator station into multiple components, but also adds additional Sentinel agents at the obstacle and sensor interfaces.

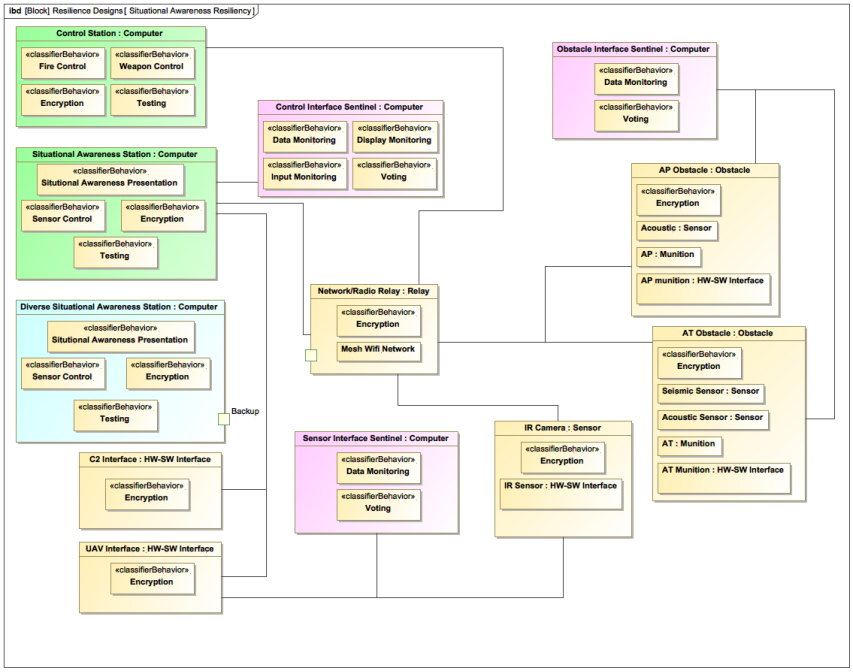

Figure 5.19 - Situational awareness resiliency architecture.

5.4 CSRM STEP 4

Step 4 involved the Red Team making assessments regarding requirements for three inter-related aspects of cyber security (Software Engineering, Cyber Defense and Cyber Attack resilience). These assessments were based upon the SE Team provided SysML descriptions related to Silverfish. While conclusions were drawn mainly based upon experience and know-how, they were supported by a review of historical cyber attack data through use of the CYBOK tool described in Section 3.2. It is important to note that the Red Team was not provided with the results derived instep 2 (Table 5.1) of the CSRM. This was done to assure that the assessment in Step 4 would principally be based upon technical factors as opposed to Blue team-derived operational factors.

The following list of conclusions were drawn by the Red Team:

1. Based upon the weapon control integrity requirements as provided in the Silverfish functional description, and assumed by the SE team in their SysMl descriptions, encryption was a desirable security requirement.

2. From a software engineering perspective, separation of the weapon control system HW/SW from the situation awareness related HW/SW, including separate operator displays, should be a design requirement. This was strongly influenced by the projected relative complexity of the situation awareness related software when compared to the control software required for weapon control.
3. Based on the highly focused weapon control sub-system functionality, use of the following SW development practices for the weapon control functions should be required:
   - Utilization of a rich suite of software quality tools (including static and dynamic test tools),
   - Extensive use of end to end testing,
   - Assembly of a high-end team of SW designers/developers focused on weapon control software development.
4. Assuming adoption of the isolation and proposed development practices for weapon control software, it was suggested that the lowest priority be assigned to a diverse redundancy resilience requirement for weapon control (i.e., resilience was less critical if the hardware/software implementation made attacks addressed to the weapon control function sufficiently difficult).
5. Suggested adoption of voice-only military communications system to higher levels of command in order to avoid potential attacks through the C2 system
6. Considered the communication sub-system as highest priority for resilience, using diverse redundancy to address attacks resulting in denial of service and message delays
7. Suggested that the resiliency design for situation awareness be the second highest priority for resilience, including diverse redundant IR sensors as a basis for addressing both reliability and cyber attack resilience requirements.
8. Suggested considering adding an operator authentication design requirement should the possibility exist for potential scenarios that require interactions across separately protected, closely located protected areas.

This set of suggested architectural and system design requirements brought with them the possibility of contention between Blue Team developed consequence-related priorities and Red Team developed cyber security priorities. Most notable was the lower resilience priority for weapon control functions suggested by the Red Team, although offset by the isolation and enhanced software engineering requirements suggested for this part of the Silverfish system. However, more generally, by introducing software engineering costs as high priorities, the affordability of cyber defense and cyber attack resilience opportunities could be impacted.

5.5 CSRM STEP 5

This step calls for the SE Team to integrate the recommendations of the Red Team into a set of corresponding SysMl representations that would be evaluated by the Blue team in preparation for management decisions regarding the architecture and preliminary design for Silverfish. As discussed in Section 5.4, the SE Team considered the disparity in prioritization related to

resilience of the weapon control subsystem to be critical. As a result of SE Team discussions, an unanticipated system architecture-related suggestion emerged; to consider providing cyber attack detection capability for the isolated weapon control system should a management decision be made to drop the resilience requirement for that part of the Silverfish system. This would include designing the system to avoid the immediate consequences of the detected cyber attack, and leaving it to the operational commanders to decide on possible non-Silverfish specific steps for continuing operations. As with other architectural and design alternatives, SysMl representations were prepared for the Step 6 CSRM meeting to expose the Blue Team to the Red Team results.

In addition, Step 5 calls for the SE Team to initiate cost analyses that would be used to inform management decisions regarding Silverfish design alternatives. Due to the limited cost analysis capabilities of the research team and the bounded scope of this project, no cost related efforts were included for the Silverfish use case.

5.6 CSRM STEP 6

This step involves gaining responses from the Blue Team regarding Red Team recommendations from Step 4 and SE Team responses developed in Step 5. Interestingly, the Blue Team supported the separation of situation awareness and weapon control functions, and was optimistic about the operators being able to use two separated displays. The Blue Team also was supportive of the Red Team's suggestion to limit communications to higher-level command to being voice only. Regarding the SE Team's Step 5 suggestions to include detection only capability for the separated weapon control function and to provide situation awareness resilience even though weapon control might not be resilient, the Blue Team suggested that resilient awareness would both support operator safety and would provide a basis for higher levels of command to take resilience-related actions that were not based upon the use of the Silverfish weapon control system functions. Finally, regarding the Red Team suggestion regarding technology-based authentication of Silverfish operators, the Blue Team would want to take a deeper look into the issue of likelihoods for closely located deployments of Silverfish that stimulated the Red Team suggestion.

CONCLUSIONS: ASSESSMENT OF RESULTS AND POTENTIAL FUTURE RESEARCH EFFORTS

This report presents a methodology for developing cyber security requirements as part of carrying out the preliminary design process for new cyber physical systems. Referred to as the Cyber Security Requirements Methodology (CSRM), it consists of a six-step process that aligns cyber security related requirements with other system requirements, and provides analysis support to allow system designers to prioritize their decisions regarding implementation of potential cyber security solutions.

Perhaps most important, this activity illuminated the point that system resilience is a system design topic, and requirements for resilience depend upon related cyber defense and software

engineering requirements. As a result, all three of these inter-related areas of cyber security should be approached concurrently, and since resilience should be addressed at the earliest phase of system design, defense and cyber security related software engineering should also be addressed at that time.

Results of the Silverfish trial application of CSRM were very promising, demonstrating that an approach for concurrently addressing resilience, defense and security related software engineering early in the system design process is feasible, and can be practically accomplished. The CSRM trial developed unanticipated results that were supported by all three of the diversely experienced teams that were engaged (operationally focused, cyber security focused and systems engineering focused). These results served to illuminate what were considered to be important system design issues that cut across operational human factors considerations, system security considerations and system software engineering considerations. However, due to the limited scope of the research effort, cost considerations could not be addressed, and as a result, no "final" decision process could be conducted based upon use of the CSRM results by program managers.

Based upon the results of this effort, it is suggested that the sponsors of the effort look into finding a weapon system development about to start, and include use of CSRM as part of that program's design process. This application would serve as a benchmark for the real-world value of CSRM, and could use the cost analysis team for the selected program to provide the needed estimates for costs.

Alternatively, if no application is feasible at this time, perhaps a weapon system technology prototyping program could employ CSRM as part of its considerations. A broader research possibility would be to embark into an exploration of a Command and Control system application that would complement the physical system application domain for CSRM. Note that earlier UVA finding regarding resiliency highlight two important aspects of weapon systems regarding resilience. First is that the risk assessment conducted via CSRM has a clearer basis for prioritization, namely human safety and weapon effectiveness. Second is that weapon systems already include safety-related design processes that serve to keep the systems more isolated and, as a result, more secure than C2 systems are.

In addition to these recommendations, the sponsors should consider using the risk analysis methodology of CSRM for cyber security related software engineering decisions. These decisions could include cost-effectiveness considerations based upon the degree of operational risk related to various components software. Those components that potentially impact the most critical operational functions can be given greater software engineering-related emphasis that those that do not. For example, the use of risk analysis as a filtering mechanism could potentially increase the productivity associated with the application of static analysis tools.

Finally, the quality and scalability of CSRM to larger system applications would require tools that can more comprehensively support cyber security related system risk analysis. These tools need to address the increasing complexity of systems, the integration of systems to support military missions (System-of-Systems), the role of humans in managing resilience in the context of the battlefield, and the desirability of rapidly importing newly available technology support arising from commercial applications. Efforts to integrate CSRM into programs that are pursuing related

tools would help to broaden the perspective of the researchers and should result in new and better tools that the DoD can utilize to enhance their opportunities to employ cyber attack resiliency solutions.

APPENDIX A: LIST OF PUBLICATIONS RESULTED

Bakirtzis, G., Carter, B. T., Fleming, C. H. and Elks, C. R. (2017) Mission aware: Evidence-based, mission-centric cybersecurity analysis. *arXiv-eprints*.

Bakirtzis, G., Carter, B. T., Fleming, C. H. and Elks, C. R. (2018) A model-based approach to security analysis for cyber-phsyical systems. *IEEE International Systems Conference*.

Carter, B. T., Bakirtzis, G., Elks, C. R. and Fleming, C. H. (2018) A systems approach for eliciting mission-centric security requirements. *IEEE International Systems Conference*.

Carter, B. T., Fleming, C. H, Elks, C. R. and Bakirtzis, G. (2018) Cyber-Physical Systems Modeling for Security Using SysML. *Conference on Systems Engineering Research (CSER)*.

Mead, N. R., Shull, F., Vemuru, K., Villadsen, O. (March 2018) A Hybrid Threat Modeling Method. Carnegie Mellon University / Software Engineering Institute Technical Note CMU/SEI-2018-TN-002.

APPENDIX B: CITED AND RELATED REFERENCES

Hause, M. (2006, September). The sysml modelling language. In *Fifteenth European Systems Engineering Conference* (Vol. 9).

Leveson, N. (2011). *Engineering a safer world: Systems thinking applied to safety*. MIT press.

Young, W., & Leveson, N. (2013, December). Systems thinking for safety and security. In *Proceedings of the 29th Annual Computer Security Applications Conference* (pp. 1-8). ACM.

Horowitz, B., Beling, P., Fleming, C., et al. (2017, December) Security Engineering – FY17 Systems Aware Cybersecurity. *Systems Engineering Research Center* Technical Report SERC-2017-TR-114.