System Requirements Specification

for

Option 9

Version 1.0 approved

Prepared by Vetronica

Embry-Riddle Aeronautical University

September 29, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Everyone	9/29/23	Initial version	Version 1.0

Everyone	10/30/23	Revisions, improvements	Version 2.0
Everyone	11/17/23	Revisions; improvements	Version 3.0
Everyone	11/21/23	Revisions; improvements	Version 3.1

1. Introduction

1.1 Purpose

The goal of this project is developing a cybersecurity planning for executing vulnerability assessment for operational environments, application of cybersecurity in operational environments and conducting decision analysis of crucial infrastructure by implementing model-based systems engineering.

1.2 Document Conventions

The purpose of this document is to give the reader an idea of our requirements for the system we plan on designing as well as the hardware, software, and user expectations. The document is divided into four different categories, each of which cover an essential part of the system.

The Overall Description section of this document will cover the basic requirements of the system, the functionalities, and the overall basic information for the system.

The External Interface Requirements section of this document will cover the interfaces. There are four different interfaces that will be covered: User Interfaces, Hardware Interfaces, Software Interfaces, and Communications Interface.

The System features section of this document will cover the key features pertinent to this system in itself. This includes importing models, analyzing the models, and rendering the results of the analysis.

Lastly, the Other Nonfunctional Requirements category just covers any requirements that were not covered in any of the above sections.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document is project owners of large computer systems which need to work despite cyber threats, as well as anyone interested in MBsE model based cyber security.

1.4 Product Scope

For this Project, the objective is to research and develop a Cybersecurity Analysis tool for use in an Aerospace Technology environment. This tool will be supported by a Model-based Systems Engineering tool to be developed by the team. By the end of the term, the expectation is that there will be a functional Cyber Analysis environment that can be used to identify safeguards in physical

and virtual aerospace environments to protect against the growing cybersecurity threats of today's world.

1.5 References

- Veronica Cyber Analysis Vision Statement
  - https://github.com/Vetronica/CybAnalysisTool/blob/main/Vetronica%20Cyber%20Analysis%20Vision%20Statement.docx
- MagicDraw System Requirements
  - https://docs.nomagic.com/display/MD190/System+requirements
- Capella System Requirements
  - https://github.com/eclipse/capella/blob/master/doc/plugins/org.polarsys.capella.ui.doc/html/Installation%20Guide/How%20to%20install%20Capella%20and%20Addons.mediawiki#table-of-contents

2. Overall Description

2.1 Product Perspective

This product is a cybersecurity analysis tool targeting aerospace applications via SysML. Using the provided MBSE tools, this will be a self-contained product which will model hardware infrastructure for obtaining a high level view of possible attack vectors.

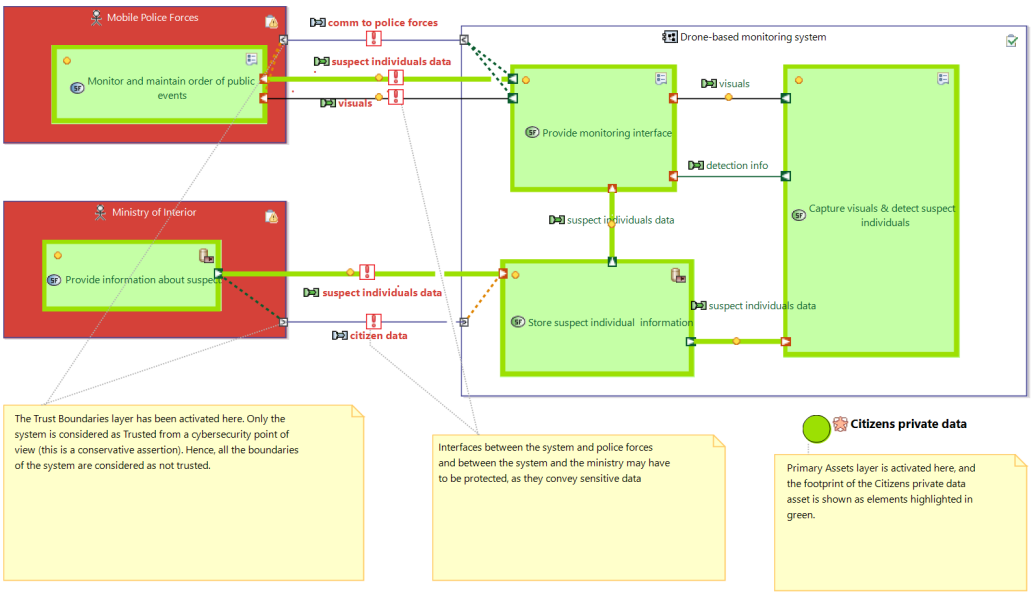

Fig 1.

2.2 Product Functions

- Define Network(s) of interest
- Define Threats of concern
- Establish applicable Conceptual data infrastructure and data structure (storage and processing)
- Establish applicable Data sphere
- Establish usage spheres
- Define cybersecurity strategies for specific threats
- Identify Data Acquisition methods (sensors, streaming, security log files)
- Identify current sensor vantage and blind spots (visual schemas)

2.3 User Classes and Characteristics

2.3.1 System Engineers

The System Engineer would be in charge of designing the system in the software, whether it be MagicDraw or Capella. This engineer will have access to all MBsE related functions, designing the system, and preparing the model for analysis for the Cybersecurity Engineers.

2.3.2 Cybersecurity Engineers

The Cybersecurity Engineers would have access to analyzing the results outputted from the system. They would be responsible for creating a Cybersecurity policy to implement whatever results were outputted, and determine whether they follow FAA regulations.

2.4 Operating Environment

This project will be primarily contained in MBSE-related modeling tools. For this project, two different MBSE tools will be used to import models: MagicDraw and Capella. These tools will be used to model a physical environment rather than the allocation of physical resources. At this juncture, the functionally of each tool and scope of integration is unknown.

2.5 Design and Implementation Constraints

Recommendations for running MagicDraw:
2.5.MD1.: Must purchase license from Dassault Systems.
2.5.MD2.: CPU of Intel Core i5 or higher
2.5.MD3.: 16 GB Ram
2.5.MD4.: 1 GB of disk space. (This could increase depending on plugins installed.)
2.5.MD5.: Display Resolution of 1920x1080
2.5.MD6.: Operating System must be compatible to run Java SE 8

Recommendations for running Capella:

2.5.C1.: 4 bit computer with Windows, Linux or Mac Operating System. (Linux and Mac have not yet been field-proven tested).
2.5.C2.: An unzip software such as 7-Zip or equivalent
2.5.C3.: 2 GHz processor
2.5.C4.: 4 GB RAM
2.5.C5.: 15 GB of available hard disk space

2.6 User Documentation

To be completed at a later date

2.7 Assumptions and Dependencies

In order to successfully use this software without any complications, the below assumptions are presented to make the experience simple.

2.7.1.: Must be working on a device compatible with the recommendations and/or requirements with both MagicDraw/Capella.
2.7.2.: Must have skill in usage of MBsE development tools such as Capella and MagicDraw.
2.7.3.: Must have an environment to be implemented through the MBsE development tool.
2.7.4.: There is a dedicated Systems Engineer and a Cybersecurity Engineer.

3. External Interface Requirements

3.1 User Interfaces

User interface is yet to be determined.

3.2 Hardware Interfaces

The hardware for this project must be capable of running both the software created here as well as the MBsE software. The interfaces for the hardware can be broken down as follows:

3.2.1.: The system is intended to be used on a modern computer running Windows, MacOs or Linux.
3.2.2.: The system shall be running one of MagicDraw or Capella to get the model.
3.2.3.: The system shall have at least 4GB of RAM.
3.2.4.: The system shall have at least 16GB of hard disk space.

3.3 Software Interfaces

Software interfaces and implementation to be determined in Sprint 3

3.4 Communications Interfaces

3.4.1.: The software will serialize the model state to disk to allow users to save their work as well as distribute finished models.

3.4.2.: The software will also check for updates from a centralized server to prompt the user if an update is available.

4. System Features

This section provides a more in-depth list of what was presented in Section 2.2. This includes the functionality on importing models (See Section 4.1), the analysis of the models (See Section 4.2), and the rendering of analysis results (See Section 4.3). This section will go over the requirements for each of these sections.

4.1 Importing of Models

4.1.1 Description and Priority

The user shall be able to import existing models from Capella or MagicDraw into the analysis tool. Separating the import process from business logic allows for flexibility in supporting both formats and improved maintainability.

4.1.2 Stimulus/Response Sequences

The user shall be able to indicate a file / directory save of a Capella or MagicDraw model for import into the analysis tool.

4.1.3 Functional Requirements

4.1.3.1: The software shall allow the user to import MagicDraw model files.
4.1.3.2: The software shall allow the user to import Capella model files.
4.1.3.3: The software shall alert the user if features in a particular save are not supported by the tool.
4.1.3.4: The software shall alert the user if a model is corrupted or cannot be loaded.
4.1.3.5: The software shall display when the model is imported successfully.

4.2 Analysis

4.2.1 Description and Priority

Once a model is imported into the software, the user will have the ability to run high level analysis on the model to uncover possible cyber threats. The exact nature of analysis depends on what concerns the user has. What networks are they interested in? What threats are they most concerned about?

4.2.2 Stimulus/Response Sequences

The user shall indicate the threat models, networks of concern via a config file (for CLI use), or by selecting options in the GUI.
The user shall indicate via a button press or subcommand that they wish to perform analysis.

4.2.3 Functional Requirements

4.2.3.1: The software shall import networks of concern from the user
4.2.3.2: The software shall import a prioritized list of threats of concern from the user

4.3 Rendering of Analysis Results

4.3.1 Description and Priority

Once a model is imported into the software, the user will have the ability to run high level analysis on the model to uncover possible cyber threats.

4.3.2 Stimulus/Response Sequences

The software shall display analysis results in a manner suitable for consumption. The exact requirements for this action are currently unknown, as they depend on if we support a CLI or GUI.

4.3.3 Functional Requirements

4.3.3.1: TBD

5. Other Nonfunctional Requirements

5.1 Performance Requirements

TBDe

5.2 Safety Requirements

Not applicable.

5.3 Security Requirements

TBD (maybe a login to make things run? OR some other security.

5.4 Software Quality Attributes

TBD

5.5 Business Rules

In a business environment, it's assumed there would be at least 2 workers, a systems engineer and a cybersecurity engineer. The systems engineer would design the model and prepare it to be analyzed in the software while the cybersecurity engineer would use those results to take any necessary action and implement necessary policies. See Section 2.3 for more information on this.

6. Other Requirements

Appendix A: Glossary

CLI - Command Line Interface
GUI - Graphical User Interface
MBsE - Model Based systems Engineering
Rust - The Rust programming language: A language empowering everyone to build reliable and efficient software.
STPA - Systems Theoretic Process Analysis

Appendix B: Analysis Models

TBD

Appendix C: To Be Determined List

3.3 - Software Interfaces
5.1 - Performance Requirements
5.3 - Security Requirements
5.4 - Software Quality Attributes