Codebook

SafeSecMergedFinalCoding_latest.mx22

7/12/2022

Code System

1 System and Its Properties	2
1.1 Domain	11
1.1.1 Automotive	13
1.1.2 UAV & other Robotics	5
1.1.3 Internet of Things	5
1.1.4 critical infrastructure & production systems	4
1.2 Self-Adaptation	2
1.2.1 Architecture	8
1.2.1.1 Mape-K or Variant	8
1.2.1.2 No Specific Architecture	2
1.2.2 Way of Implementation	7
1.2.2.1 Adaptation Limitation	5
1.2.2.2 Adaptation Strategy	0
1.2.2.2.1 Increased Vigilance	0
1.2.2.2.1.1 Increase Security Level	1
1.2.2.2.1.2 Increase Safety Margin	1
1.2.2.2.1.3 Block Communication	4
1.2.2.2.2 Stop Operation	0
1.2.2.2.2.1 Restart System	4
1.2.2.2.2.2 End Cooperation	6
1.2.2.2.2.3 Stop Operation	2
1.2.2.2.2.4 Return to Base	3
1.2.2.2.3 Not Specified	4
1.2.2.2.4 Service Reduction	0
1.2.2.2.4.1 Degeneracy	1
1.2.2.2.4.2 Minimal Operation	1
1.2.2.2.4.3 Save Energy, Computation Time	1
1.2.2.2.4.4 Remove Component	4
1.2.2.2.5 Redundancy	2
1.2.2.2.5.1 State Estimation	3
1.2.2.2.5.2 Redundant Sensor	2
1.2.2.2.5.3 Redundant Controller	6
1.2.2.2.6 Other	7
1.2.2.2.6.1 Grant Priviledges	3

1.2.2.2.6.2 Code Decryption	1
1.2.2.3 Adaptation Realization	0
1.2.2.3.1 Adaptation Definition	1
1.2.2.3.2 Execute	6
1.2.2.3.3 Plan	6
1.2.2.3.4 Monitor	8
1.2.2.3.5 Analyze	8
1.2.2.3.6 Adaptation Verification	1
1.2.2.3.7 Adaptation Components	5
1.2.3 Attack on Self-Adaptation	3
1.3 Degree of Automation	0
1.3.1 Manual	1
1.3.2 Semi-automated	12
1.3.3 Fully-automated	13
2 Integration	0
2.1 Security Attacks	0
2.1.1 CIA	0
2.1.1.1 Confidentiality	12
2.1.1.2 Integrity	18
2.1.1.3 Availability	15
2.1.2 Attack Surface	0
2.1.2.1 Remote Access	17
2.1.2.2 Physical Access	1
2.1.2.3 Close Proximity	3
2.1.3 Attack Mechanisms	34
2.1.3.1 Manipulate Data Structures	0
2.1.3.2 Employ Probabilistic Techniques	0
2.1.3.3 Manipulate Timing and State	0
2.1.3.4 Abuse Existing Functionality	2
2.1.3.5 Collect and Analyze Information	4
2.1.3.6 Subvert Access Control	6
2.1.3.7 Inject Unexpected Items	10
2.1.3.8 Engage in Deceptive Interactions	11
2.1.3.9 Manipulate System Resources	10
2.1.4 Type of Data	0
2.1.4.1 Risk Indicators	6
2.1.4.2 Threat	6

2.1.4.3 Vulnerability	3
2.1.4.4 Asset	14
2.1.5 How is Severity Measured	0
2.1.5.1 Continous	1
2.1.5.2 Discrete	2
2.1.6 Affected Part of Adaptation	1
2.1.6.1 Database	1
2.1.6.2 Controller	7
2.1.6.3 Sensors	7
2.2 Safety Hazards	0
2.2.1 Hazard Cause	0
2.2.1.1 Other	5
2.2.1.2 String Stability Decrease	2
2.2.1.3 Crash	4
2.2.1.4 Component Failure	3
2.2.1.5 Overheating	2
2.2.1.6 System Failure	6
2.2.1.7 Performance Alteration	5
2.2.1.8 Collision	6
2.2.2 Hazard Source	0
2.2.2.1 External	17
2.2.2.2 Internal	7
2.2.3 Hazard Cause Old (Deprecated)	0
2.2.3.1 Collision with Another Object	5
2.2.3.2 Environmental Conditions	2
2.2.3.3 Hardware Failure	10
2.2.3.4 External attack	17
2.2.4 Safety Quality Factors	0
2.2.4.1 Environmental	3
2.2.4.2 Property	12
2.2.4.3 Health	12
2.3 Integration of Security and Safety	0
2.3.1 Loosely Integrated	13
2.3.2 Fully Integrated	8
3 Modeling Approach	6
3.1 Model@Runtime	5
3.2 Context of the Model	0

3.2.1 Modeling the System	0
3.2.1.1 State and behavior	15
3.2.1.2 For illustration purposes	2
3.2.1.3 State	3
3.2.1.4 Behavior	15
3.2.2 Adaptation	3
3.2.3 Security	42
3.2.4 Safety	19
3.2.5 Other	0
3.3 Meta-Model Type / Modeling Language	0
3.3.1 Logic	7
3.3.2 Functional (Fault/Attack Tree)	8
3.3.3 Structural (Component Diagram)	2
3.3.4 Behavioral (Automata)	5
3.3.4.1 Mathematical Model	26
3.3.5 Other	2
3.4 Analysis Objectives	0
3.4.1 Detection	12
3.4.2 Requirements	20
3.4.3 Risk and Threat Analysis	12
3.4.4 Adaptation	3
3.4.5 Evaluation	30
3.4.6 Other	16
3.5 Adaptation Property Checked	1
3.5.1 Stability	2
3.5.2 Accuracy	1
3.5.3 Settling Time	1
3.5.4 Small Overshoot	0
3.5.5 Robustness	1
3.5.6 Termination	1
3.5.7 Consistency	1
3.5.8 Scalability	1
3.5.9 Security	4
3.5.10 Dependability	1
3.5.11 String Stability	2
3.5.12 Controllability	2
3.5.13 Observation	0

3.5.14 Safety	4
3.5.15 Node Criticality Index	1
3.6 Verification and Validation Technique	0
3.6.1 Testing	12
3.6.2 Formal Verification	4
4 Standards	0
4.1 Security	0
4.1.1 ISO/IEC 9126	1
4.1.2 ISO/SAE 21434	2
4.1.3 ISO/IEC18045	1
4.2 Safety	0
4.2.1 IEC 61508	1
4.2.2 ISO 31000:2009	1
4.2.3 ISO 26262	6
4.3 Agnostic	0
4.3.1 ISO/IEC 15026-2	0
4.3.2 SAE J3016	4
5 Challenges	0
5.1 Addressed	0
5.1.1 Verification	1
5.1.1.1 Addressed by	1
5.1.2 Adaptation	1
5.1.2.1 Addressed by	2
5.1.3 Attack	3
5.1.3.1 Addressed by	4
5.1.4 Detecting Defects	4
5.1.4.1 Addressed by	4
5.1.5 Survivability and Resilience	5
5.1.5.1 Addressed by	6
5.1.6 System and Environment	8
5.1.6.1 Addressed by	12
5.1.7 Other	2
5.1.7.1 Addressed by	3
5.2 Open	0
5.2.1 Modeling	0
5.2.2 System	4
5.2.3 Environment	5

5.2.4 Safety and Security	4
5.2.5 Adaptation	7
5.2.6 Other	1
6 Treatment	2
6.1 Hazard Elimination	6
6.2 Hazard Reduction	8
6.3 Hazard Control	7
6.4 Hazard Damage Minimization	0
7 Other	1

1 System and Its Properties

1.1 System and Its Properties >> Domain

This is the broad domain that the self-adaptive system is used in. An approach can apply to multiple domains.

1.1.1 System and Its Properties >> Domain >> Automotive

The automotive domain contains systems consisting mainly of manned road vehicles e.g. autonomous cars, fleet and platoon management or connected cars and infrastructure.

1.1.2 System and Its Properties >> Domain >> UAV & other Robotics

The unmanned aerial vehicles domain includes drones (multicopter, unmanned planes or helicopter) and their ground control stations. Other robotics include industrial robots, logistics robots etc.

1.1.3 System and Its Properties >> Domain >> Internet of Things

The Internet of Things domain contains (systems of) systems consisting mainly of low power, sensor augmented, "intelligent" connected devices

1.1.4 System and Its Properties >> Domain >> critical infrastructure & production systems

Critical infrastructure and production includes large scale adaptive systems, e.g. industry 4.0 factory lines, power plants etc.

1.2 System and Its Properties >> Self-Adaptation

1.2.1 System and Its Properties >> Self-Adaptation >> Architecture

Type specifies the type of architecture the adaptation system is based on.

1.2.1.1 System and Its Properties >> Self-Adaptation >> Architecture >> Mape-K or Variant

MAPE-K architectures or variants use one or more adaptation cycles that **M**onitor the system state, **A**nalyze it to detect unwanted states, **P**lan an adaptation e.g. to minimize risk, and **E**xecute this adaptation using clearly separated components that may share a common **K**nowledge base

1.2.1.2 System and Its Properties >> Self-Adaptation >> Architecture >> No Specific Architecture

1.2.2 System and Its Properties >> Self-Adaptation >> Way of Implementation

details of the implementation of the system adaptation

1.2.2.1 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Limitation

limitations to the adaptation that are mentioned, e.g. cases the adaptation fail in

1.2.2.2 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy

descriptions of the general strategy or goal of the system adaptation, e.g. getting to a failsafe state or trying to resolve the problem

1.2.2.2.1 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Increased Vigilance

trying to avoid safety or security problems by enforcing a higher safety/security level, e.g. adapting firewall rules to be more strict after an attempted intrusion to the system was detected

1.2.2.2.1.1 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Increased Vigilance >> Increase Security Level

1.2.2.2.1.2 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Increased Vigilance >> Increase Safety Margin

1.2.2.2.1.3 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Increased Vigilance >> Block Communication

1.2.2.2.2 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Stop Operation

stopping operation, ending cooperation, shutting down or other ways to reach a failsafe state that also ends normal operation

1.2.2.2.2.1 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Stop Operation >> Restart System

1.2.2.2.2.2 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Stop Operation >> End Cooperation

1.2.2.2.2.3 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Stop Operation >> Stop Operation

1.2.2.2.2.4 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Stop Operation >> Return to Base

1.2.2.2.3 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Not Specified

it is described that the system adapts, but not in which way

1.2.2.2.4 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Service Reduction

reduce service, e.g. to decrease the attack surface, that limits but not stops normal operation, e.g. switch to a worse performing but safer algorithm

1.2.2.2.4.1 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Service Reduction >> Degeneracy

1.2.2.2.4.2 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Service Reduction >> Minimal Operation

1.2.2.2.4.3 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Service Reduction >> Save Energy, Computation Time

1.2.2.2.4.4 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Service Reduction >> Remove Component

1.2.2.2.5 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Redundancy

switch to a redundant system, sensor, or algorithm that generally can do the same but is e.g. not affected by an security problem

1.2.2.2.5.1 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Redundancy >> State Estimation

1.2.2.2.5.2 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Redundancy >> Redundant Sensor

1.2.2.2.5.3 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Redundancy >> Redundant Controller

1.2.2.2.6 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Other

other strategies that are not listed

1.2.2.2.6.1 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Other >> Grant Priviledges

1.2.2.2.6.2 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Strategy >> Other >> Code Decryption

1.2.2.3 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Realization

how the adaptation is technically realized

1.2.2.3.1 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Realization >> Adaptation Definition

a definition of what an adaptation is in the context of the paper (e.g. architectural reconfiguration or parameter change)

1.2.2.3.2 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Realization >> Execute

details on the execute part of the MAPE-K cycle

1.2.2.3.3 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Realization >> Plan

details on the plan part of the MAPE-K cycle

1.2.2.3.4 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Realization >> Monitor

details on the monitoring part of the MAPE-K cycle

1.2.2.3.5 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Realization >> Analyze

details on the analysis part of the MAPE-K cycle

1.2.2.3.6 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Realization >> Adaptation Verification

description of a system that verifies whether the adaptation succeeded

1.2.2.3.7 System and Its Properties >> Self-Adaptation >> Way of Implementation >> Adaptation Realization >> Adaptation Components

details on the component structure and component interaction of the adaptation system

1.2.3 System and Its Properties >> Self-Adaptation >> Attack on Self-Adaptation

description of possible attacks on the adaptation system

1.3 System and Its Properties >> Degree of Automation

degree of manual input needed from a user to trigger and execute an adaptation

1.3.1 System and Its Properties >> Degree of Automation >> Manual

the adaptation or parts of it must be done manually

1.3.2 System and Its Properties >> Degree of Automation >> Semi-automated

the adaptation is semi-automated, e.g. an adaptation plan is proposed by the system and the user has to accept it

1.3.3 System and Its Properties >> Degree of Automation >> Fully-automated

the adaptation happens automatically without user input at the time of the adaptation (but maybe earlier to specify possible adaptations)

2 Integration

2.1 Integration >> Security Attacks

2.1.1 Integration >> Security Attacks >> CIA

Confidentiality, integrity and availability, also known as the CIA triad, is a model designed to guide policies for information security within an organization.

2.1.1.1 Integration >> Security Attacks >> CIA >> Confidentiality

Confidentiality is roughly equivalent to Confidentiality measures are designed to prevent sensitive information from unauthorized access attempts. It is common for data to be categorized according to the amount and type of damage that could be done if it fell into the wrong hands. More or less stringent measures can then be implemented according to those categories.

2.1.1.2 Integration >> Security Attacks >> CIA >> Integrity

Integrity involves maintaining the consistency, accuracy and trustworthiness of data over its entire lifecycle. Data must not be changed in transit, and steps must be taken to ensure data cannot be altered by unauthorized people (for example, in a breach of confidentiality).

2.1.1.3 Integration >> Security Attacks >> CIA >> Availability

Availability means information should be consistently and readily accessible for authorized parties. This involves properly maintaining hardware and technical infrastructure and systems that hold and display the information.

2.1.2 Integration >> Security Attacks >> Attack Surface

Attacks can be classified based on the initial entry point, which includes physical, close proximity and remote access.

2.1.2.1 Integration >> Security Attacks >> Attack Surface >> Remote Access

The remote access involves attacks that are implemented over large distances over the network that utilize GPS, radio, and the internet.

2.1.2.2 Integration >> Security Attacks >> Attack Surface >> Physical Access

The physical access represents the direct access to wires and control boxes, usually achieved by an on-site attack.

2.1.2.3 Integration >> Security Attacks >> Attack Surface >> Close Proximity

The close proximity access includes the attacks that focus on communication with the system such as sensor, audio, and dedicated short-range communication attacks.

2.1.3 Integration >> Security Attacks >> Attack Mechanisms

The Common Attack Pattern Enumeration and Classification (CAPEC) scheme postulated by the MITRE Corporation demonstrates a hierarchical framework of the different ways by which systems are attacked by a foreign entity. These techniques are: Deceptive Interactions, Abuse of Existing Functionality of the System, Data Structure Manipulation, System Resources Manipulation, Injection of Unexpected Items, Employing Probabilistic Techniques, Manipulation of System Timing and State, Collect and Analyze Information, and Circumventing or Subverting Access Control.

2.1.3.1 Integration >> Security Attacks >> Attack Mechanisms >> Manipulate Data Structures

Attack patterns in this category manipulate and exploit characteristics of system data structures in order to violate the intended usage and protections of these structures.

2.1.3.2 Integration >> Security Attacks >> Attack Mechanisms >> Employ Probabilistic Techniques

An attacker utilizes probabilistic techniques to explore and overcome security properties of the target that are based on an assumption of strength due to the extremely low mathematical probability that an attacker would be able to identify and exploit the very rare specific conditions under which those security properties do not hold.

2.1.3.3 Integration >> Security Attacks >> Attack Mechanisms >> Manipulate Timing and State

An attacker exploits weaknesses in timing or state maintaining functions to perform actions that would otherwise be prevented by the execution flow of the target code and processes.

2.1.3.4 Integration >> Security Attacks >> Attack Mechanisms >> Abuse Existing Functionality

An adversary uses or manipulates one or more functions of an application in order to achieve a malicious objective not originally intended by the application, or to deplete a resource to the point that the target's functionality is affected.

2.1.3.5 Integration >> Security Attacks >> Attack Mechanisms >> Collect and Analyze Information

Attack patterns within this category focus on the gathering, collection, and theft of information by an adversary. The adversary may collect this information through a variety of methods including active querying as well as passive observation.

2.1.3.6 Integration >> Security Attacks >> Attack Mechanisms >> Subvert Access Control

An attacker actively targets exploitation of weaknesses, limitations and assumptions in the mechanisms a target utilizes to manage identity and authentication as well as manage access to its resources or authorize functionality.

2.1.3.7 Integration >> Security Attacks >> Attack Mechanisms >> Inject Unexpected Items

Attack patterns within this category focus on the ability to control or disrupt the behavior of a target either through crafted data submitted via an interface for data input, or the installation and execution of malicious code on the target system.

2.1.3.8 Integration >> Security Attacks >> Attack Mechanisms >> Engage in Deceptive Interactions

Attack patterns within this category focus on malicious interactions with a target in an attempt to deceive the target and convince the target that it is interacting with some other principal and as such take actions based on the level of trust that exists between the target and the other principal.

2.1.3.9 Integration >> Security Attacks >> Attack Mechanisms >> Manipulate System Resources

Attack patterns within this category focus on the adversary's ability to manipulate one or more resources in order to achieve a desired outcome.

2.1.4 Integration >> Security Attacks >> Type of Data

The type of data dimension is significant because it provides information regarding what security data is being used in attacks that were described in the investigated set of papers.

2.1.4.1 Integration >> Security Attacks >> Type of Data >> Risk Indicators

This code demonstrates if the risk data was used within the investigated approaches.

2.1.4.2 Integration >> Security Attacks >> Type of Data >> Threat

This code demonstrates if the threat data was used within the investigated approaches.

2.1.4.3 Integration >> Security Attacks >> Type of Data >> Vulnerability

This code demonstrates if the vulnerability data was used within the investigated approaches.

2.1.4.4 Integration >> Security Attacks >> Type of Data >> Asset

This code demonstrates if the asset data was used within the investigated approaches.

2.1.5 Integration >> Security Attacks >> How is Severity Measured

This dimensions indicates if the approaches provide any information regarding how the severity of attacks is measured.

2.1.5.1 Integration >> Security Attacks >> How is Severity Measured >> Continous

This code show if the continous metrics were used to measure severity.

2.1.5.2 Integration >> Security Attacks >> How is Severity Measured >> Discrete

This code show if the discrete metrics were used to measure severity.

2.1.6 Integration >> Security Attacks >> Affected Part of Adaptation

This dimension shows if the attack targets a specific component related to adaptation mechanisms.

2.1.6.1 Integration >> Security Attacks >> Affected Part of Adaptation >> Database

This code is used if the attack target is the database of the self-adaptive system.

2.1.6.2 Integration >> Security Attacks >> Affected Part of Adaptation >> Controller

This code is used if the attack target is the controller of the self-adaptive system.

2.1.6.3 Integration >> Security Attacks >> Affected Part of Adaptation >> Sensors

This code is used if the attack target are the sensors that interact with the environment and are directly related to the self-adaptive system.

2.2 Integration >> Safety Hazards

2.2.1 Integration >> Safety Hazards >> Hazard Cause

Hazard cause represents the potential reasons behind why the safety property was affected. This can occur due to the following reasons: collision with another object, environmental conditions, hardware failure, loss of control, pilot error and external attack.

2.2.1.1 Integration >> Safety Hazards >> Hazard Cause >> Other

Hazards that do not belong to any of the main categories are part of the "Other" category.

2.2.1.2 Integration >> Safety Hazards >> Hazard Cause >> String Stability Decrease

String stability effects platoons or swarms. With an increased number of participants there is potential for more errors which

2.2.1.3 Integration >> Safety Hazards >> Hazard Cause >> Crash

Hazards that result in a crash of a vehicle, drone, etc. This means that the vehicle is damaged to a deegree that it could potentially endanger driver or passengers.

2.2.1.4 Integration >> Safety Hazards >> Hazard Cause >> Component Failure

Component Failure occurs when a single component or part of the system is unresponsive.

2.2.1.5 Integration >> Safety Hazards >> Hazard Cause >> Overheating

Overheating occurs when a system's temperature is higher compared to the maximum allowed temperature.

2.2.1.6 Integration >> Safety Hazards >> Hazard Cause >> System Failure

System failure occurs when a whole system becomes unresponsive due to a hazard.

2.2.1.7 Integration >> Safety Hazards >> Hazard Cause >> Performance Alteration

This hazard results in performance increase or decrease of a system. For example, this can be speed of a vehicle.

2.2.1.8 Integration >> Safety Hazards >> Hazard Cause >> Collision

Colission is a hazard between vehicle and other vehicle or any other object in which the health of the passenger is not endangered.

2.2.2 Integration >> Safety Hazards >> Hazard Source

Hazard source dimension shows the potential reasons behind the hazard occurence. Hazard source can be classified as internal and external.

2.2.2.1 Integration >> Safety Hazards >> Hazard Source >> External

This code includes all the external hazard sources that are not related to the system itself.

2.2.2.2 Integration >> Safety Hazards >> Hazard Source >> Internal

This code includes all the hazard sources that are internal and could relate to the system itself.

2.2.3 Integration >> Safety Hazards >> Hazard Cause Old (Deprecated)

Hazard cause represents the potential reasons behind why the safety property was affected. This can occur due to the following reasons: collision with another object, environmental conditions, hardware failure, loss of control, pilot error and external attack.

2.2.3.1 Integration >> Safety Hazards >> Hazard Cause Old (Deprecated) >> Collision with Another Object

This code includes hazard causes in which the reason is the collision between two objects such as a crash of two vehicles.

2.2.3.2 Integration >> Safety Hazards >> Hazard Cause Old (Deprecated) >> Environmental Conditions

This code includes hazard causes in which the reason is the environment such as a bad weather condition.

2.2.3.3 Integration >> Safety Hazards >> Hazard Cause Old (Deprecated) >> Hardware Failure

This code includes hazard causes in which the reason is failure of the system itself.

2.2.3.4 Integration >> Safety Hazards >> Hazard Cause Old (Deprecated) >> External attack

This code includes hazard causes in which the reason is an external attack. This could be an attacker trying to penetrate the system.

2.2.4 Integration >> Safety Hazards >> Safety Quality Factors

In the case that a hazard occurs, it is important to know what factors were affected. This includes health, property and environmental factors.

2.2.4.1 Integration >> Safety Hazards >> Safety Quality Factors >> Environmental

This code demonstrates if a certain hazard has any effect on the environment. For example, if the vehicle damages any surrounding environmental object, that would be an affect on environment.

2.2.4.2 Integration >> Safety Hazards >> Safety Quality Factors >> Property

This code demonstrates if a certain hazard has any effect on the property. For example, if the crash damages the vehicle itself, this would be an effect on the property.

2.2.4.3 Integration >> Safety Hazards >> Safety Quality Factors >> Health

This code demonstrates if a certain hazard has any effect on the health. For instance, in a case of a vehicle crash, it is possible that a person within the car is injured, which would be an effect on health.

2.3 Integration >> Integration of Security and Safety

This dimension shows if certain approaches consider integration of security and safety.

2.3.1 Integration >> Integration of Security and Safety >> Loosely Integrated

This code indicates if certain approach considers integration of security and safety, but to a very low deegree.

2.3.2 Integration >> Integration of Security and Safety >> Fully Integrated

This code indicates if certain approach considers integration of security and safety, but to a high deegree.

3 Modeling Approach

This code contains all the subcodes that are used to answer RQ2 and that are relevant in terms of modeling a self-adaptive system, as well as its safety and security aspects.
This code is also applied to text passages that are interesting in the context of modeling but do not fit any subcode.

3.1 Modeling Approach >> Model@Runtime

This code is applied to text passages that state that a model is used at runtime.

3.2 Modeling Approach >> Context of the Model

This code contains all the subcodes that describe a context in which a model is used.

3.2.1 Modeling Approach >> Context of the Model >> Modeling the System

This code contains all the subcodes that describe the usage of models to present a system.

3.2.1.1 Modeling Approach >> Context of the Model >> Modeling the System >> State and behavior

This code is applied to text passages that describes a mathematical model that is used as basis for simulations and therefore describe the state and the behavior of a system.

3.2.1.2 Modeling Approach >> Context of the Model >> Modeling the System >> For illustration purposes

This code is applied to text passages that relate to a model that is only used to illustrate the system described in a paper.

3.2.1.3 Modeling Approach >> Context of the Model >> Modeling the System >> State

This code is applied to text passages that describe the usage of a model to represent a state of a system.

3.2.1.4 Modeling Approach >> Context of the Model >> Modeling the System >> Behavior

This code is applied to text passages that describe the usage of a model to represent the behavior of a system.

3.2.2 Modeling Approach >> Context of the Model >> Adaptation

This code is applied to text passages that describe the use of models in the context of adaptation. For example, models that are used to describe an adaptation or used to reason about adaptation.

3.2.3 Modeling Approach >> Context of the Model >> Security

This code is applied to text passages that describe the use of models in the context of security. For example, a typical model used in the context of security is an attack tree.

3.2.4 Modeling Approach >> Context of the Model >> Safety

This code is applied to text passages that describe the use of models in the context of safety. This includes, for example, Boolean expressions, which are used to express safety conditions.

3.2.5 Modeling Approach >> Context of the Model >> Other

This code is applied to text passages that describe the use of models in another context than the system, adaptation, security or safety.

3.3 Modeling Approach >> Meta-Model Type / Modeling Language

This code contains all the subcodes that describe the type of the models used.

3.3.1 Modeling Approach >> Meta-Model Type / Modeling Language >> Logic

This code is applied to text passages that mention a logic-based modelling language, e.g., binary decision diagrams.

3.3.2 Modeling Approach >> Meta-Model Type / Modeling Language >> Functional (Fault/Attack Tree)

This code is applied to text passages that mention models describing a dynamic process.

3.3.3 Modeling Approach >> Meta-Model Type / Modeling Language >> Structural (Component Diagram)

This code is applied to text passages that mention models describing the static structure of a system.

3.3.4 Modeling Approach >> Meta-Model Type / Modeling Language >> Behavioral (Automata)

This code is applied to text passages that mention models describing the dynamics of a system.

3.3.4.1 Modeling Approach >> Meta-Model Type / Modeling Language >> Behavioral (Automata) >> Mathematical Model

This code is applied to text passages that mention the usage of mathematical models to describe the dynamics of a system.

3.3.5 Modeling Approach >> Meta-Model Type / Modeling Language >> Other

This code is applied to text passages in which models are mentioned that cannot be assigned to any of the other categories.

3.4 Modeling Approach >> Analysis Objectives

This code contains all the subcodes that are used to further categorize the different objectives of the analyses.

3.4.1 Modeling Approach >> Analysis Objectives >> Detection

This code is applied to text passages that mention analyses to detect anomalies.

3.4.2 Modeling Approach >> Analysis Objectives >> Requirements

This code is applied to text passages that mention analyses to check whether given requirements are met.

3.4.3 Modeling Approach >> Analysis Objectives >> Risk and Threat Analysis

This code is applied to text passages that mention risk or threat analysis.

3.4.4 Modeling Approach >> Analysis Objectives >> Adaptation

This code is applied to text passages that mention analyses that focus on the adaptation of a system.

3.4.5 Modeling Approach >> Analysis Objectives >> Evaluation

This code is applied to text passages that mention analyses that focus on the evaluation of a system.

3.4.6 Modeling Approach >> Analysis Objectives >> Other

This code is applied to text passages in which analyses are mentioned that cannot be assigned to any of the other categories.

3.5 Modeling Approach >> Adaptation Property Checked

These are the different non-functional properties that are checked, verified or validated to determine if and when adaptation needs to happen.

3.5.1 Modeling Approach >> Adaptation Property Checked >> Stability

This is a property related to the local stability of a vehicle within a platoon. When the degree of perturbation in a vehicle in a platoon decreases with time, the system is stable (locally stable).

3.5.2 Modeling Approach >> Adaptation Property Checked >> Accuracy

This is a measure of the correctness of of the behaviour of a system within a bounded limit of deadline misses.

3.5.3 Modeling Approach >> Adaptation Property Checked >> Settling Time

I aggregate recovery time to mean the same as settling time. This is based my understanding of the definition of settling time here: *Santiago Hurtado, Sagar Sen, Rubby Casallas. Reusing Legacy Software in a Self-adaptive Middleware Framework. Adaptive and Relfective Middleware Workshop, Middleware 2011, Dec 2011, Lisbon, Portugal. 2011.*

3.5.4 Modeling Approach >> Adaptation Property Checked >> Small Overshoot

how much a disturbance makes system go beyond its expected behaviour.

3.5.5 Modeling Approach >> Adaptation Property Checked >> Robustness

a property of redundancy within the system

3.5.6 Modeling Approach >> Adaptation Property Checked >> Termination

A function to mitigate against delay in communication or malicious attack.

3.5.7 Modeling Approach >> Adaptation Property Checked >> Consistency

Is consistency == continuity == integrity?

VirtualDrone paper

3.5.8 Modeling Approach >> Adaptation Property Checked >> Scalability

a measure of the potential impact of an attack or foreign influence on a system

3.5.9 Modeling Approach >> Adaptation Property Checked >> Security

the ability of a system to protect data and information, as well as prevent unauthorized access to such system remaining attack resilient.

3.5.10 Modeling Approach >> Adaptation Property Checked >> Dependability

An encompassing property for an available, reliable and integruous system

3.5.11 Modeling Approach >> Adaptation Property Checked >> String Stability

ability of a vehicle platoon to obey a constant control law.

3.5.12 Modeling Approach >> Adaptation Property Checked >> Controllability

a measure of the impact of a driver on security in an autonomous system.

3.5.13 Modeling Approach >> Adaptation Property Checked >> Observation

a metric to consider safety without human control and to predict failure countermeasures

3.5.14 Modeling Approach >> Adaptation Property Checked >> Safety

a measurement/metric for safety violations

3.5.15 Modeling Approach >> Adaptation Property Checked >> Node Criticality Index

relates to safety, security and quality of service

3.6 Modeling Approach >> Verification and Validation Technique

These are the various assurance strategies for ensuring that system behaves according to laid down requirements.

3.6.1 Modeling Approach >> Verification and Validation Technique >> Testing

Testing includes simulation, *hardware-in-the-loop* (HITL), *software-in-the-loop* (SITL), code analysis, ML based testing etc.

3.6.2 Modeling Approach >> Verification and Validation Technique >> Formal Verification

Formal verification includes the use of model checking, temporal logic, program synthesis techniques etc.

4 Standards

This dimension shows if certain standards were addressed within the investigated approaches.

4.1 Standards >> Security

This dimension shows if security standards were addressed within the investigated approaches.

4.1.1 Standards >> Security >> ISO/IEC 9126

ISO/IEC 9126 is an international standard proposed to make sure **'quality of all software – intensive products'** .

4.1.2 Standards >> Security >> ISO/SAE 21434

It specifies engineering requirements for cybersecurity risk management regarding concept, product development, production, operation, maintenance and decommissioning of electrical and electronic (E/E) systems in road vehicles, including their components and interfaces.

4.1.3 Standards >> Security >> ISO/IEC18045

ISO/IEC 18045:2008 is a companion document to ISO/IEC 15408, *Information technology - Security techniques - Evaluation criteria for IT security*. ISO/IEC 18045:2008 defines the minimum actions to be performed by an evaluator in order to conduct an ISO/IEC 15408 evaluation, using the criteria and evaluation evidence defined in ISO/IEC 15408. ISO/IEC 18045:2008 does not define evaluator actions for certain high assurance ISO/IEC 15408 components, where there is as yet no generally agreed guidance.

4.2 Standards >> Safety

This dimension shows if safety standards were addressed within the investigated approaches.

4.2.1 Standards >> Safety >> IEC 61508

EC 61508-1:2010 covers those aspects to be considered when electrical/electronic/programmable electronic (E/E/PE) systems are used to carry out safety functions. A major objective of this standard is to facilitate the development of product and application sector international standards by the technical committees responsible for the product or application sector. This will allow all the relevant factors, associated with the product or application, to be fully taken into account and thereby meet the specific needs of users of the product and the application sector. A second objective of this standard is to enable the development of E/E/PE safety-related systems where product or applic

4.2.2 Standards >> Safety >> ISO 31000:2009

SO 31000:2009 provides principles and generic guidelines on risk management.

ISO 31000:2009 can be used by any public, private or community enterprise, association, group or individual. Therefore, ISO 31000:2009 is not specific to any industry or sector.

4.2.3 Standards >> Safety >> ISO 26262

ISO 26262 is intended to be applied to safety-related systems that include one or more electrical and/or electronic (E/E) systems and that are installed in series production passenger cars with a maximum gross vehicle mass up to 3 500 kg. ISO 26262 does not address unique E/E systems in special purpose vehicles such as vehicles designed for drivers with disabilities.

4.3 Standards >> Agnostic

This dimension shows if other standards aside from security and safety were addressed within the investigated approaches.

4.3.1 Standards >> Agnostic >> ISO/IEC 15026-2

ISO/IEC 15026-2:2011 specifies minimum requirements for the structure and contents of an assurance case to improve the consistency and comparability of assurance cases and to facilitate stakeholder communications, engineering decisions, and other uses of assurance cases.

4.3.2 Standards >> Agnostic >> SAE J3016

The SAE J3016:2021 standard defines terminology for automated vehicles including the famous SAE Automation Levels. It is widely referenced in discussions, other standards, and even government regulations. Unfortunately, what is said about J3016 is too often inaccurate, misleading, or just plain incorrect.

5 Challenges

This code contains all the subcodes to classify addressed and open challenges in the context of safety and security of self-adaptive systems.

5.1 Challenges >> Addressed

This code contains all the subcodes that describe the origin of addressed challenges.

5.1.1 Challenges >> Addressed >> Verification

This code is applied to text passages that describe the addressed challenges which emerge from verification a system.

5.1.1.1 Challenges >> Addressed >> Verification >> Addressed by

This code is applied to text passages that describe how a challenge, which emerge from verification a system, is addressed.

5.1.2 Challenges >> Addressed >> Adaptation

This code is applied to text passages that describe the addressed challenges which emerge from the adaptive behavior of a system.

5.1.2.1 Challenges >> Addressed >> Adaptation >> Addressed by

This code is applied to text passages that describe how a challenge, which emerge from the adaptive behavior of a system, is addressed.

5.1.3 Challenges >> Addressed >> Attack

This code is applied to text passages that describe the addressed challenges which emerge from attacks leading to safety issues.

5.1.3.1 Challenges >> Addressed >> Attack >> Addressed by

This code is applied to text passages that describe how a challenge, which emerge from attacks leading to safety issues, is addressed.

5.1.4 Challenges >> Addressed >> Detecting Defects

This code is applied to text passages that describe the challenges addressed that emerge when detecting defects.

5.1.4.1 Challenges >> Addressed >> Detecting Defects >> Addressed by

This code is applied to text passages that describe how a challenge, which emerge when detecting defects, is addressed.

5.1.5 Challenges >> Addressed >> Survivability and Resilience

This code is applied to text passages that describe the addressed challenges which emerge from improving the survivability or resilience of a system.

5.1.5.1 Challenges >> Addressed >> Survivability and Resilience >> Addressed by

This code is applied to text passages that describe how a challenge, which emerge from improving the survivability or resilience of a system, is addressed.

5.1.6 Challenges >> Addressed >> System and Environment

This code is applied to text passages that describe the addressed challenges which emerge from the type of the system or its environment.

5.1.6.1 Challenges >> Addressed >> System and Environment >> Addressed by

This code is applied to text passages that describe how a challenge, which emerge from the type of the system or its environment, is addressed.

5.1.7 Challenges >> Addressed >> Other

This code is applied to text passages that describe the addressed challenges which emerge from other origins.

5.1.7.1 Challenges >> Addressed >> Other >> Addressed by

This code is applied to text passages that describe how a challenge, which emerge from other origins, is addressed.

5.2 Challenges >> Open

This code contains all the subcodes that describe the origin of open challenges.

5.2.1 Challenges >> Open >> Modeling

This code is applied to text passages that describe the open challenges which emerge from the usage of models.

5.2.2 Challenges >> Open >> System

This code is applied to text passages that describe the open challenges which emerge from the type of the system.

5.2.3 Challenges >> Open >> Environment

This code is applied to text passages that describe the open challenges which emerge from the environment of a system.

5.2.4 Challenges >> Open >> Safety and Security

This code is applied to text passages that describe the open challenges which emerge from the combined consideration of safety and security of a system.

5.2.5 Challenges >> Open >> Adaptation

This code is applied to text passages that describe the open challenges which emerge from the adaptive behavior of a system.

5.2.6 Challenges >> Open >> Other

This code is applied to text passages that describe the open challenges which emerge from other origins.

6 Treatment

How the different attacks or changes in the system are handled

6.1 Treatment >> Hazard Elimination

a strategy to forestall any hazard

6.2 Treatment >> Hazard Reduction

helps to decrease the hazard caused by attacks or changes in the system

6.3 Treatment >> Hazard Control

technique to reduce exposure to hazard by substitutive mechanisms

6.4 Treatment >> Hazard Damage Minimization

technique to remain available even after impact has be made on the system

7 Other