# Model Driven Engineering for Resilience of Systems with Black Box and AI-based Components

**Conference Paper** · January 2022

DOI: 10.1109/RAMS51457.2022.9893930



**5 authors**, including:

Nikolaos Papakonstantinou
VTT Technical Research Centre of Finland
**81** PUBLICATIONS **1,045** CITATIONS
SEE PROFILE

Joonas Linnosmaa
VTT Technical Research Centre of Finland
**27** PUBLICATIONS **173** CITATIONS
SEE PROFILE

Douglas Lee Van Bossuyt
Naval Postgraduate School
**135** PUBLICATIONS **937** CITATIONS
SEE PROFILE

# Model Driven Engineering for Resilience of Systems with Black Box and AI-based Components

Nikolaos Papakonstantinou[1], Britta Hale, Joonas Linnosmaa, Jarno Salonen and Douglas L. Van Bossuyt

Key Words: Safety, Security, Model Driven Engineering, Resilience, Defense in Depth, AI, Black Box Components

## *INTRODUCTION*

Modern complex cyber-physical systems heavily rely on humans and AI for mission-critical operations and decision making. Unfortunately, these components are often "black boxes" to the operator, either because the decision models are too complex for human comprehension (e.g. deep neural networks) or are intentionally hidden (e.g. proprietary intellectual property). In these cases, the decision logic cannot be validated and therefore trust is forced.

Development of system modeling techniques for past influences when data/internals of specific critical components cannot be accessed is a challenge, as is the case with AI and human components. This is a recognized contemporary concern for industrial operators and government agencies, since stakeholders of large engineering projects typically do not want to share design data or model access. From the client point of view, there is a need for modeling system resilience (safety and security) when there is lack of complete trust/control in the AI process or over human factors and fail-safes - layers of defense should be deployed.

Prior work has presented a methodology for assessing and supporting development of resilience for mission critical systems that include AI components and humans. Zero Trust and Defense-in-Depth (DiD) principles within the methodology protect critical components, taking into account interfaces and influences during different lifecycle phases and system configurations. However, the methodology does not cover analysis of past influences of critical components which is a very important but laborious task that could be supported by model driven engineering methods.

This work extends that of prior methodologies and presents ways to systematically model the past influences for critical human and AI components. The concept is based on metamodels for interaction/dependency modeling and then on the definition of metrics in order to establish and even reduce the search space across past influences, and add controls when a search path is not followed.

## *1 LITERATURE REVIEW*

### 1.1 *Combining safety and security towards resilience engineering*

The demanding complexity of modern systems requires the parallel development of safety engineering methods and tools applicable to all lifecycle phases (specification, design, operation, maintenance, decommission, etc.). Domain specific safety standards like the more than 200 published standards for nuclear safety from the International Atomic Energy Agency (IAEA) [1] contain a rigorous (and challenging) framework of requirements and recommendations. Safety assessment methods like the Failure Modes and Effects Analysis (FMEA) [2] Probabilistic Risk Assessment (PRA) [3] can guide practitioners to systematically analyze and assess the safety a system based on the potential failure modes of the system components, the failure propagation paths based on the system topology and the predicted consequences. Although in safety engineering unknown/unexpected failures/consequences are possible, engineers can take advantage of a wealth of past knowledge and statistical data to calculate the overall risk (within error margins). This contrasts the uncertainty and more dynamic nature of security engineering [4].

As with safety, security is both critical to a system and complicated to assess under increased system complexity. As complexity grows, so do the subtleties in potential system vulnerabilities. Two secure components can even be naïvely combined to build an insecure system since what constitutes a cyberattack on component pieces may differ from the whole. For example, a camera sensor component may be "secure" if it the device internals are resistant to tampering and a decision component may be "secure" if the decision processes and outcomes are authenticated. However, if these are naïvely combined without a secure channel, an adversary could inject or manipulate data en route between the sensor and decision element, leading to second-order effects as outcomes of the system. Security assessment must cover a wide variety of cyber attributes, including networks, software, algorithms, and user access control. Design and assessment current methodologies are different within each sub-domain, tailored to sub-domain security needs and it is vital to ensure that countermeasures within one subdomain are not mistaken as a replacement for security within others. The DevSecOps approach has gained traction in software development, for example, as an integration

[1] Corresponding author.

of security, development and operations into a unified and efficient software development lifecycle [5]. Unfortunately, such methods address only a restricted slice of the cybersecurity domain; e.g. it is possible to securely develop insecure software using insecure underlying algorithms, or securely develop software that interacts insecurely with other software. Efforts have been made to more comprehensively address system security, such as with the NIST Risk Management Framework (RMF) [6]. While it has been applied extensively, NIST's RMF has also faced criticism [7, 8] due to the its inability to keep pace with emerging security technologies and adversarial abilities as well as the relative ease in which an insecure system can pass RMF checkboxes due to developer error arising from malicious intent, ignorance, or oversight. Thus, for any given system, security assessment must take into account the security components (e.g. networking, software, algorithms, etc.) and how they are combined.

Safety and security share a common goal, to protect the system under study from disturbances. Despite that, safety and security are often handled separately during design and operation. Approaches combining safety and security engineering are still active areas of research, at least in complex systems such as the nuclear domain [9]. *Resilience* is a system property that combines safety and security, it is defined as "The intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances so that it can sustain required operations under both expected and unexpected conditions" [10]. Expected and unexpected conditions need to be anticipated, monitored, mitigated and be a source of knowledge/experience for the future. A concept part of resilience is *survivability*, "the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents" [11], regardless of the nature of the disruption (probabilistic failure or malicious attack) the system should sustain vital functions to satisfy its mission. Survivability refers to both physical failures and attacks as well as cyber failures and attacks.

There is similarity between the relation of reliability with safety and the relation of security with security technologies. Safety assumes that anything can potentially fail and critical systems system must therefore be designed to handle the impact of common cause failures [12]. Following this mindset, in addition to hardening the security of individual components/software/algorithms/networks/etc., it is also important to design the security of the overall system so that a cyberattack on any individual component is not catastrophic to the entire system. Zero Trust [13] is an architecture paradigm that supports this goal by focusing on perimeter-less designs and the establishment of trust in an ad-hoc manner for all system elements (including human resources), e.g. trust is never assumed based on a previously achieved user clearance/authentication nor on the location of a component in a network/zone. Zero Trust can be combined with the basic principles of Defense-in-Depth [14] like *redundancy* (use of additional components/systems beyond the bare minimum needed for a function) and *diversity* (use of components with different technologies or operation principles to implement the redundancy) to design systems with increased resilience [15]. Controls based on redundancy and diversity can be put in place to increase the confidence that no single system component can compromise the system (data/function/mission) even if access or control is obtained by a malicious actor.

### 1.2 *Model Driven Engineering for resilience*

Model Driven Engineering (MDE) or Model Based System Engineering (MBSE) [16] is a well established system engineering paradigm where throughout the lifecycle of a system (specification, design, operation, maintenance, decommission) calls for the development of system models covering system aspects like topology/behavior for each engineering disciplines and lifecycle phases as well the dependencies between them. The goal is to improve the handling of system complexity through traceability and hierarchical abstraction as well as to identify emergent behavior [17]. The Unified Modelling Language (UML) [18] with its

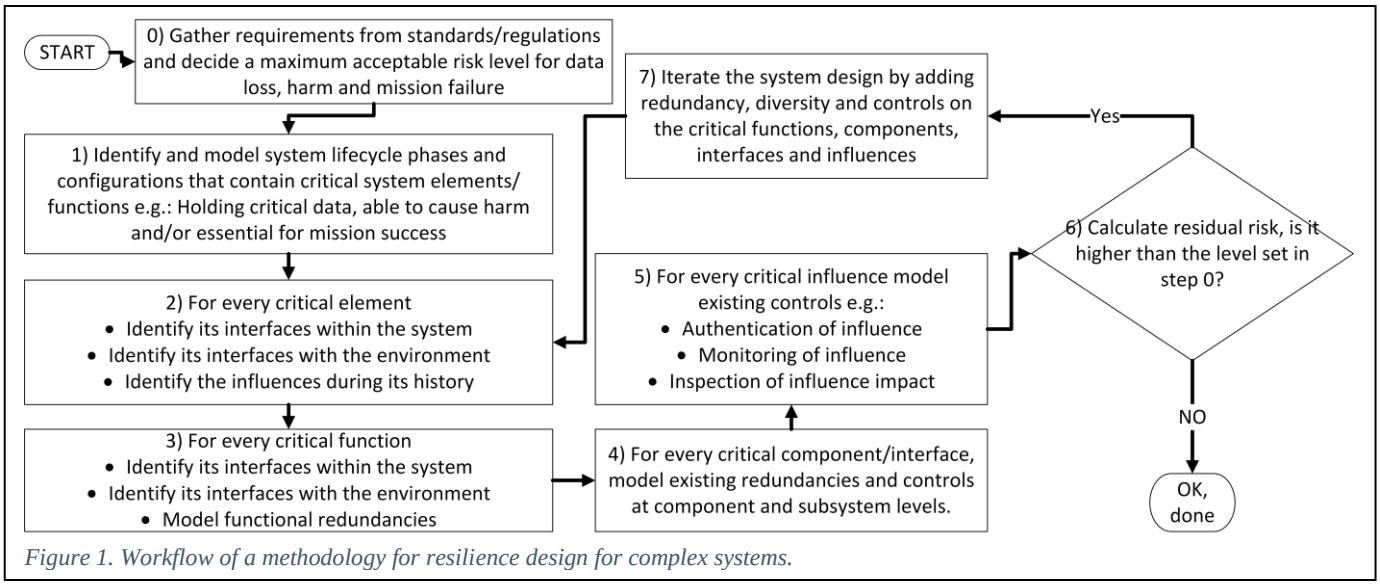


*Figure 1. Workflow of a methodology for resilience design for complex systems.*

simple to use and customize class diagrams is enough for basic system modelling, although for mode detailed modelling the Systems Modelling Language (SysML) is a better choice [19]. System models have been exploited in past work for safety assessment like the automatic generation of Fault Tree and Event Tree models [20, 21] as well as the evaluation of DiD and Zero Trust principles [22, 23].

Part of MDE is the creation of functional models that describe the high level decomposition of complex systems [24]. Research has shown that functional models can be useful for the early identification of failure propagation paths [25] as well as the development of more efficient AI-based fault detection and identification systems [26]. In this paper the functional model provides the means for designing systems to withstand the loss of a function regardless of the specific component-level failure/attack.

## 2 METHODOLOGY

This paper extends the methodology for system resilience engineering presented in previous work [22] with a modelling framework aiming to support the practitioner to identify the critical system functions, components, interfaces and influences.

### 2.1 Methodology overview - workflow

Previous work presented a workflow for the systematic analysis of critical systems for the identification of weaknesses abased on the Zero Trust principle (i.e. nothing can be trusted without control/redundancy) [22]. The methodology has been slightly refined, see Fig. 1. The basic steps are the setup of the basic goals of the resilience of the system in terms of the maximum acceptable of risk in categories like loss of data, loss of mission and causing of harm (Step 0).

The first step of the methodology includes the creation of system component and functional models for the lifecycle phases where there is potential risk. These models can be basic dependency models [27] with the key system components and environment elements. In this implementation UML class diagram [18] models were utilized, but ideally the real engineering models can be used instead to avoid re-work and reduce the error probability. Critical system elements and functions are identified as the ones whose loss can impact the resilience of the system.

Steps 2 and 3 call for listing the internal/external interfaces and influences on the critical components and functions while steps 4 and 5 iterate through the list of interfaces/influences to establish the existence of Zero Trust - DiD controls.

Step 6 is the calculation of the overall risk for the categories of step 0 and if it is lower than the threshold then the process ends. If the risk is higher in one or more categories, step 7 calls for additional controls mitigations to be added to the design and the method iterates to calculate the new risk levels.

### 2.2 Simple metamodel for resilience modelling

To enable the modelling work needed to support the methodology of this paper, a simple metamodel with the basic concepts was developed as a UML profile, basically a collection of UML "Stereotypes" [18]. This profile can be used to customize class diagram models with the specific concepts needed to perform the modelling described in the methodology. The profile diagram is shown in Fig. 2, there are nine stereotypes that extend the "Class" UML element through the "RiskProbabilities" which adds the properties needed to hold safety and security related rick estimations for events directly affecting the elements or for the propagation of failures/attacks. element, these are the:

- "SystemElement" and "ExternalElement" used for modelling the system components and environment
- "Internal and external interface" of components and functions
- "Influence" for modelling influences on system components
- "Control" for modelling implemented controls on components and functions

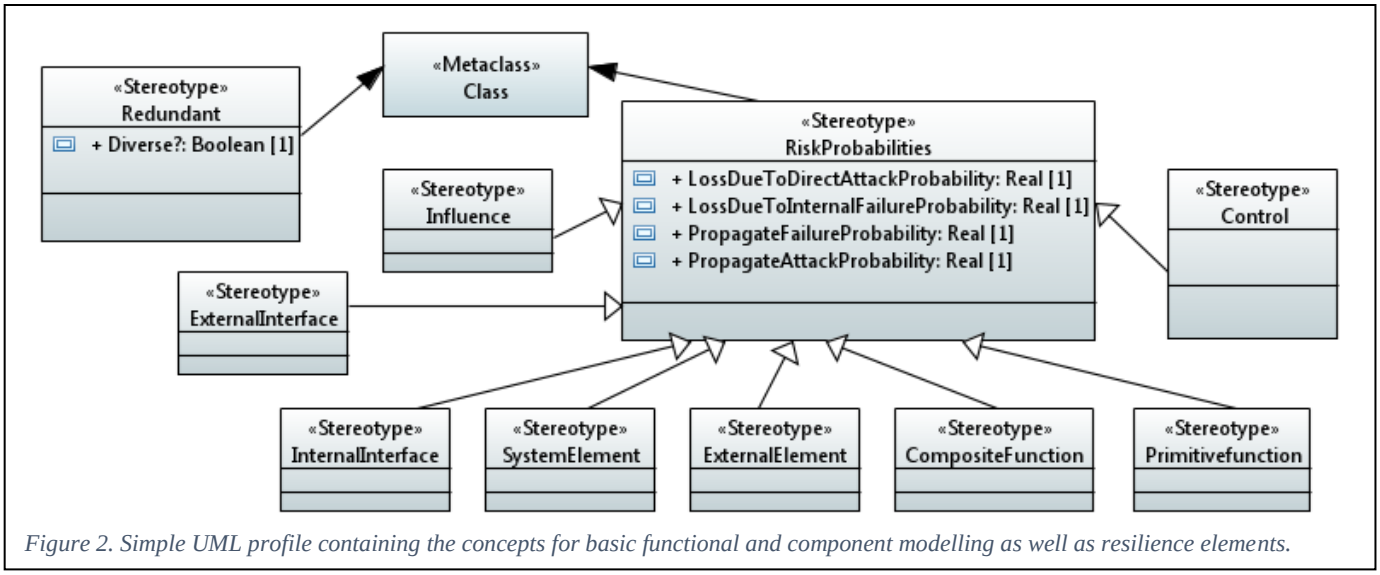


*Figure 2. Simple UML profile containing the concepts for basic functional and component modelling as well as resilience elements.*

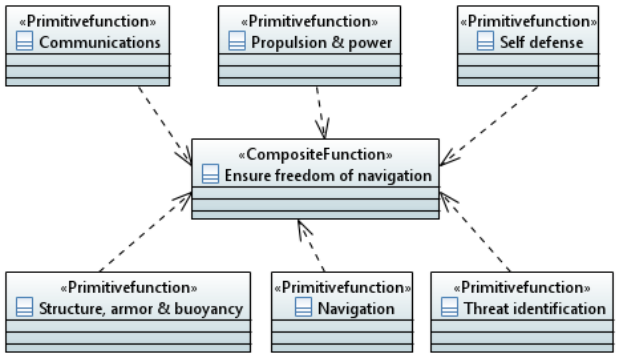


*Figure 3. Functional model of the USV case study.*

- "CompositeFunction" and "PrimitiveFunction" to be able to create hierarchical functional models

Additionally the "Redundant" directly extends the "Class" UML element. This stereotype can be applied to model elements to capture the concept of redundant functions and system elements with an option for diversity.

## 3 CASE STUDY

### 3.1 Case study description

The modelling aspect of the methodology is demonstrated on the case study of a fictional autonomous Unmanned Surface Vessel (USV) developed to conduct freedom of navigation missions in contested maritime environments. The USV's mission is to patrol a pre-determined area and in a remote controlled or an autonomous mode defend against threats.

Critical functions of the system, whose loss due to fault or attack at least affects mission success are:

1) Structure, armor & buoyancy – Propulsion & power; failure can also lead to the USV being vulnerable to additional attacks and possible data/technology extraction.
2) Navigation; failure can also lead to the USV entering prohibited waters and/or being captured causing intelligence/technology extraction. This function is AI-based (when the USV is in autonomous mode).
3) Threat identification - Self defense; failure can also lead to harm of non-combatant/civilian actors in the operations theater or to capture of the USV. This function is AI-based (when the USV is in autonomous mode).
4) Communications; failure can also lead to the hijacking of the USV opening possibilities for harming civilians, provocation, capturing the USV leading to loss of intelligence/technology.

In critical functions enabled by AI (e.g. Navigation, Threat identification), we take a black-box methodology approach. After training due to the complexity of the AI-based decision models it is not possible to be verify that their operation is correct (as expected) for all possible inputs. This leads to vulnerabilities during training and later on in the lifecycle of these functions.

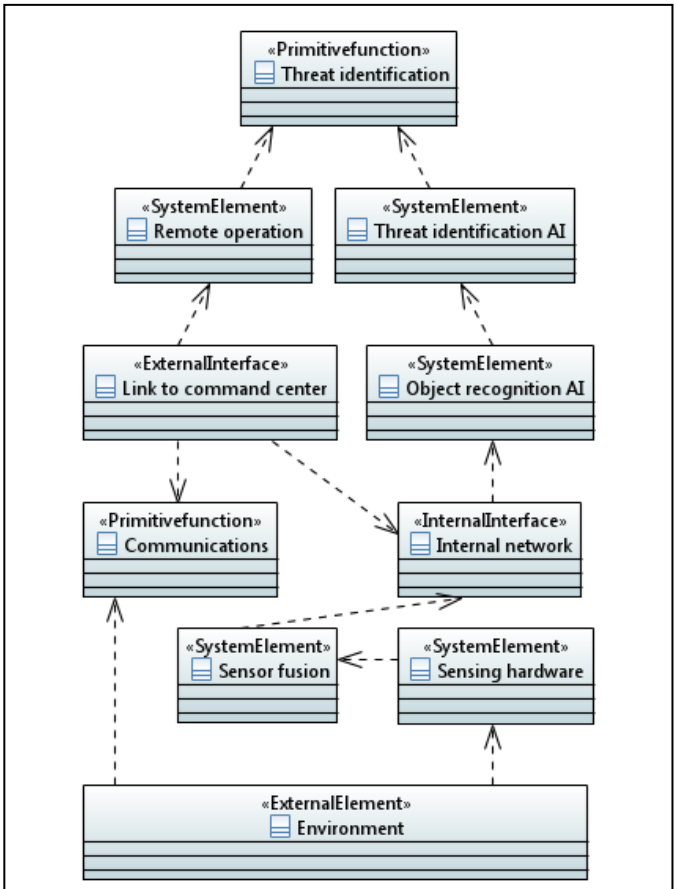


*Figure 4. Component model for the Threat identification function*

### 3.2 Case study modelling for resilience

The modelling concepts presented in section 2.2 are applied to the case study to support the application of the resilience methodology presented in section 2.1.

For this example, the focus is on the risk of causing harm to civilians. A certain threshold is set by standards and/or other means. It is determined that one of the lifecycle phase that contains such risk is during operation – patrolling of waters. The modelling focuses on this phase.

A simple functional model of the system is presented in Fig. 3. It contains the high level function of the USV during operation (perform mission - endure freedom of navigation) which is decomposed to 6 primitive functions (as described in section 3.1).

As one of the AI-enabled functions affecting the probability of harming civilians, the "Threat identification" function will be analyzed further for this case study. A simplified and partial model of the components linked to this function is shown in Fig. 4. The function is based on sensing that is processed and then feeds and object recognition and a threat identification module. A common bus is enabling the internal communication as well as access to the interface to the remote operator.

A partial model of the past influences on the AI-enabled components is shown in Fig. 5. The engineers responsible for

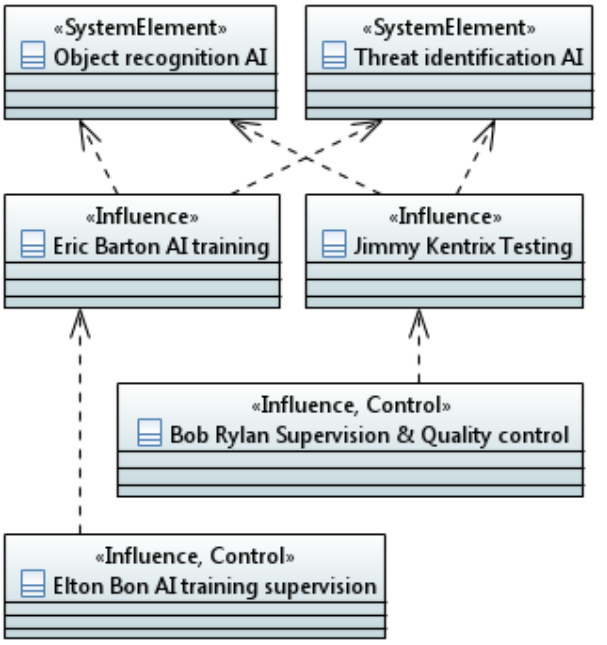


*Figure 5. Past influences on AI-enabled components*

training and testing of the AI components are modelled. An issue is that they appear to work alone (although in more realistic projects, these tasks are done by teams). Another issue is that the training and the testing engineer are involved in the development of both AI components, thus increasing their influence to the system. As a first response, two more supervising engineers are added to reduce the risk.

Fig. 6 shows the redundancies and controls added to the system to reduce the overall risk. At a functional level there can be a redundant function for threat identification alongside a control. Redundancies and associated controls can be implemented for the critical components and interfaces.

## *SUMMARY & CONCLUSIONS*

The modelling framework proposed serves as a demonstrator - proof of concept. Ideally, the resilience modelling concepts should be integrated to the computer aided engineering tools, so that there is no need to re-model parts of the system (even at a higher level). The dependency model needed for the resilience assessment could, in principle, be automatically extracted from the different engineering diagrams, the engineering/maintenance databases, product lifecycle management tools, organization maps, etc.

It is important to note that the proposed method and workflow can cover both external threat vectors and potential biases inherent in the AI model itself.

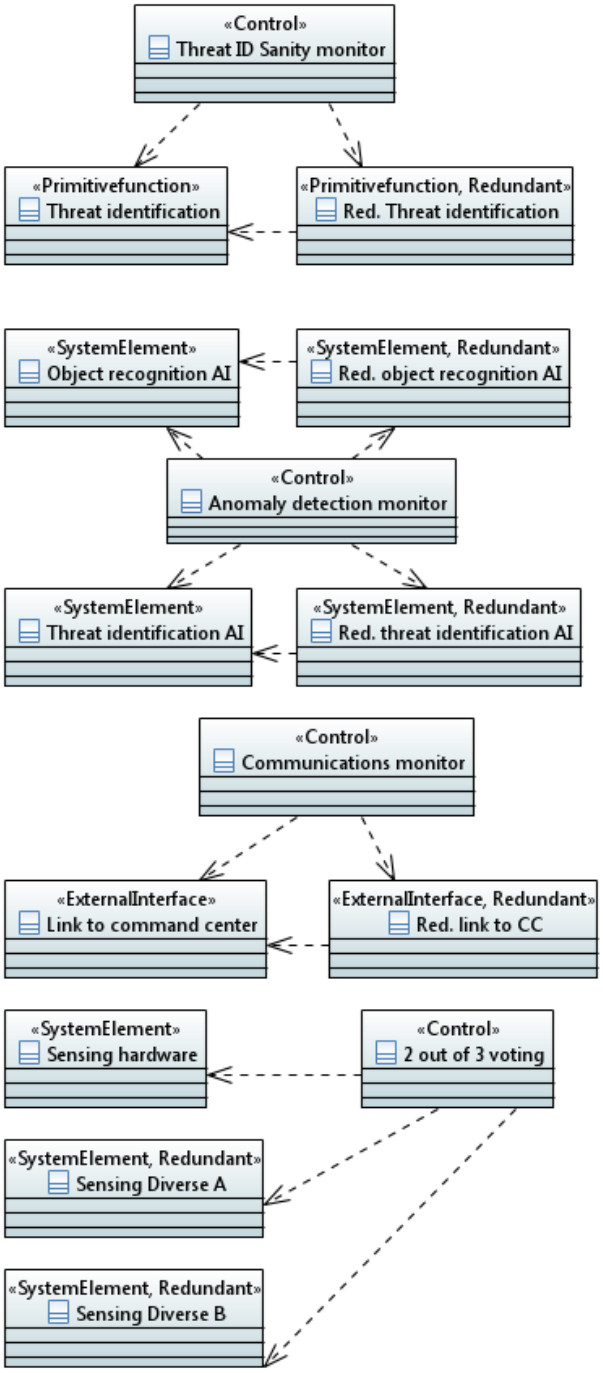


*Figure 6. Function, component and interface redundancies and controls*

## *REFERENCES*

[1] IAEA. (2021). International Safety Standards (IAEA). Available: https://www.onr.org.uk/iaea.htm

[2] I. E. Commission, "IEC 60812:2006 - Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA)," ed, 2006.

[3] M. Stamatelatos and G. Apostolakis, Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners: NASA, Safety and Mission Assurance, 2002.

[4] M. B. Line, O. Nordland, L. Røstad, and I. A. Tøndel, "Safety vs. Security? (PSAM-0148)", PSAM, M. G. Stamatelatos and H. S. Blackman, Eds., ed: ASME Press, 2006, p. 0.

[5] M. Sánchez-Gordón and R. Colomo-Palacios, "A Multivocal Literature Review on the use of DevOps for e-Learning systems,"

presented at the Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 2018.
[6] NIST. (2021). NIST Risk Management Framework RMF. Available: https://csrc.nist.gov/projects/risk-management/about-rmf
[7] C. Jackson, B. Cowles, and S. Russell, "Beyond the Beltway - The Problems with NIST's Approaches to Cybersecurity and Alternatives for NSF Science," presented at the 2017 NSF Cybersecurity Summit, Virginia, USA, 2017.
[8] D. Maclean, "The NIST Risk Management Framework: Problems and recommendations," Cyber Security: A Peer-Reviewed Journal, vol. 1, 1st of December 2017 2017.
[9] IAEA, "International Atomic Energy Agency, Computer Security of Instrumentation, and Control Systems at Nuclear Facilities, IAEA Nuclear Security Series, No. 33-T," ed. Vienna, 2018.
[10] E. Hollnagel, "How Resilient Is Your Organisation? An Introduction to the Resilience Analysis Grid (RAG)," presented at the Sustainable Transformation: Building a Resilient Organization, Toronto, Canada, 2010.
[11] D. Fisher, R. Linger, H. Lipson, T. Longstaff, N. Mead, and R. Ellison, "Survivable Network Systems: An Emerging Discipline, (CMU/SEI-97-TR-013)," Software Engineering Institute, Carnegie Mellon University1997.
[12] IAEA. (1992). Procedures for Conducting Common Cause Failure Analysis in Probabilistic Safety Assessment. Available: https://www.iaea.org/publications/908/procedures-for-conducting-common-cause-failure-analysis-in-probabilistic-safety-assessment
[13] S. Rose, Oliver Borchest, Stu Mitchell, and Sean Connelly, "Zero Trust Architecture (2nd Draft) SP 800-207 (Draft)," NIST2020.
[14] IAEA, Defence in Depth in Nuclear Safety. Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 1996.
[15] N. Papakonstantinou, J. Linnosmaa, A. Z. Bashir, T. Malm, and D. L. Van Bossuyt, "Early combined safety - security Defense in Depth assessment of complex systems," RAMS 2020, Palm Springs, CA, USA, 2020.
[16] S. Friedenthal, R. Griego, and M. Sampson, "INCOSE Model Based Systems Engineering (MBSE) Initiative," presented at the INCOSE 2007 Symposium, San Diego, California, USA, 2007.
[17] L. Berardinelli, A. Mazak, O. Alt, M. Wimmer, and G. Kappel, "Model-Driven Systems Engineering: Principles and Application in the CPPS Domain," in Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects, S. Biffl, A. Lüder, and D. Gerhard, Eds., ed Cham: Springer, 2017, pp. 261-299.
[18] Object Management Group (OMG). (2015). OMG Unified Modeling Language (OMG UML) specification. Available: http://www.omg.org/spec/UML/
[19] Object Management Group (OMG). (2012). Object Management Group Systems Modeling Language (OMG SysML).
[20] N. Papakonstantinou, J. Linnosmaa, J. Alanen, and B. O'Halloran, "Automatic Fault Tree Generation from Multidisciplinary Dependency Models for Early Failure Propagation Assessment," presented at the ASME 2018 International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE), Quebec City, Canada, 2018.
[21] N. Papakonstantinou, S. Sierla, B. O'Halloran, and I. Y. Tumer, "A Simulation Based Approach to Automate Event Tree Generation for Early Complex System Designs", ASME IDETC/CIE 2013, Portland, Oregon, USA, 2013.
[22] B. Hale, D. L. Van Bossuyt, N. Papakonstantinou, and B. O'Halloran, "A Zero-Trust Methodology for Security of Complex Systems with Machine Learning Components," presented at the ASME 2021 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), Virtual, USA, 2021.
[23] N. Papakonstantinou, T. Tommila, B. O'Halloran, J. Alanen, and D. V. Bossuyt, "A Model Driven Approach for Early Assessment of Defense in Depth Capabilities of Complex Sociotechnical Systems," ASME IDETC/CIE 2017, Cleveland, Ohio, USA, 2017.
[24] R. Stone and K. Wood, "Development of a Functional Basis for Design," Journal of Mechanical Design, vol. 122, pp. 359-370, 2000.
[25] T. Kurtoglu, I. Y. Tumer, and D. Jensen, "A functional failure reasoning methodology for evaluation of conceptual system architectures," Research in Engineering Design, vol. 21, 2010.
[26] N. Papakonstantinou, S. Proper, B. O'Halloran, and I. Y. Tumer, "A Plant-Wide and Function-Specific Hierarchical Functional Fault Detection and Identification (HFFDI) System for Multiple Fault Scenarios on Complex Systems," ASME 2015 IDETC/CIE, Boston, Massachusetts, USA, 2015.
[27] A. Qamar, C. J. J. Paredis, J. Wikander, and C. During, "Dependency Modeling and Model Management in Mechatronic Design," Journal of Computing and Information Science in Engineering, vol. 12, pp. 041009-041009, 2012.

## *BIOGRAPHIES*

**Nikolaos Papakonstantinou**, D.Sc. (Tech.), Docent
VTT Technical Research Centre of Finland
e-mail: nikolaos.papakonstantinou@vtt.fi
Dr. Papakonstantinou is a Research Team Leader focusing on industrial cybersecurity topics.

**Britta Hale**, PhD
Naval Postgraduate School, Monterey, CA, USA
e-mail: britta.hale@nps.edu
Dr. Hale is a cryptographer and Assistant Professor in Computer Science at the Naval Postgraduate School. She focuses on applied cryptography & communications security.

**Joonas Linnosmaa**, M.Sc. (Tech.)
VTT Technical Research Centre of Finland
e-mail: joonas.linnosmaa@vtt.fi
M.Sc. Linnosmaa is a Research Scientist focusing on complex system resilience and data driven methods.

**Jarno Salonen**, M.Sc. (Tech.)
VTT Technical Research Centre of Finland
e-mail: jarno.salonen@vtt.fi
M.Sc. Salonen is a senior researcher and experienced project manager focusing on industrial cybersecurity.

**Douglas L. Van Bossuyt**, PhD
Naval Postgraduate School, Monterey, CA, USA
e-mail: douglas.vanbossuyt@nps.edu
Dr. Van Bossuyt is an assistant professor in the Systems Engineering Department at the Naval Postgraduate School where he focuses on the design and assessment of complex systems.