Increasingly-Autonomous CPS: Taming Emergent Behaviors from an Architectural Perspective

Jerome Hugues[1,*,†], Daniela Cancila[2,†]

[1]Carnegie Mellon University, Software Engineering Institute
[2]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

Abstract
The safety demonstration of Increasingly-Autonomous Cyber-Physical Systems is posing new challenges for the safety community: standards and practices must be adjusted to account for the system's new capabilities and operations. In this position paper, we advocate for consideration of the CPS architecture in both its functional and non-functional dimensions as a cornerstone for safety assessment. We discuss challenges to support our claim.

Keywords
Increasingly-Autonomous Cyber-Physical Systems, Resilience, Architecture

1. Introduction

Increasingly-Autonomous Cyber-Physical Systems (IA-CPS for short) are emerging as the natural evolution of embedded real-time systems[1]. The first generations of embedded systems were basic control loops operating over a self-repeating cycle. Growth in computational power and sensor capabilities lead to several evolutions, from deterministic to optimal controllers, and then Artificial Intelligence (AI) functions built around Markov decision process, machine learning, deep neural networks, etc. Hence, IA-CPS have a complex architecture that weaves hardware, AI-enabled or decision-making processes, human operators, and safety-critical software. They are time-sensitive and substitute human actions with high-frequency real-time algorithms. Their architecture involves more coupling between multiple data flows, and is more prone to timing or data corruption/bias cascading errors.

Generally speaking, IA-CPS depend on fault mitigation mechanisms correctly integrated into a functional architecture to fulfill their mission. If not so integrated, safety mechanisms can play an adversarial role and create emerging faulty behaviors with hardware, AI-enabled functionality, human operators, and the system architecture itself as fault sources. Recent incidents involving CPS at large, e.g. autonomous vehicles, are posing new challenges both from an engineering and a safety evaluation perspective [2, 3]. A general concern is that, if the safety of operations does not meet expectations, human operators will eventually distrust the system.

In this position paper, we advocate that the safety assessment of IA-CPS requires a careful review of the coupling between AI functions (e.g. image classification, decision-making processes) and the architecture of the CPS that hosts it. In section 2 we introduce the general context of safety for IA-CPS. In section 3 we introduce the general issues. In section 4 we illustrate how emergent behaviors arise at the AI/CPS boundary. In section 5 we discuss resilience assurance and we provide some research direction we are doing in our respective institutions. Finally, in section 6 we provide the conclusion.

2. On Safety Assessment and Faults in IA-CPS

Current safety standards (such as MIL-STD882 for military systems, ARP4761 for avionics, or ISO26262 for automotive) and more generally the existing body of practice usually consider faults as a conjunction of basic events, attached to their probability of occurrence. This led to successful applications in the safety-critical industry. Yet, these standards do not apply to IA-CPS, as the following premises are invalid or insufficient for complex AI-based systems:

- A Human operator can act as the final judge to control the system: the system can reach a safe state (space, nuclear, train) or the operator can

The IJCAI-ECAI-22 Workshop on Artificial Intelligence Safety (AISafety 2022), July 24-25, 2022, Vienna, Austria
*Corresponding author.
†These authors contributed equally.
jhugues@andrew.cmu.edu (J. Hugues); daniela.cancila@cea.fr (D. Cancila)
0000-0003-0148-7175 (J. Hugues); 0000-0002-3483-7947 (D. Cancila)
© 2022 Copyright 2022 Carnegie Mellon University and Daniela Cancila. This material is based upon work funded and supported in part by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

take over close control of the system (aircraft, car). This assumption is no longer true for IA-CPS: the pace of action or the size of the system state space outweighs human capacity.
- Confidence in the system safety stems from the application of rigorous safety standards. However, current standards do not consider a high level of autonomy, having embedded artificial intelligence, and consider only functions defined through requirements engineering. Yet, most IA functions are not defined by explicit requirements but rather from training data sets [4].
- Each function is fully characterized by a finite set of requirements, and the system is validated against them. Instead, autonomous artificial intelligence based functions are grey boxes [5]. One can demonstrate general properties of the system, but validation relies on incomplete simulation or tests [6].

One path to improve the state of practice is to increase trust in AI functions [7], in other words to improve confidence that an AI function is either correct or at least resilient to some faults. In [8, 9], the authors propose a fault taxonomy that discusses the class of faults and when they are most likely to appear in the system lifecycle, yet they focus mostly on the engineering of AI functions and associated activities. They do not contemplate the system as a whole.

This approach is incomplete. First, the functions of an IA-CPS cannot be fully characterized; second, they operate in an environment that is sampled by a network of sensors, processors, and software functions. Thus, we advocate for an alternative approach: we consider faults in IA-CPS as emergent behavior that a system must resist or eventually control.

3. Emergent Behaviors and System's Architecture

Sifakis and al. define the emergent properties of a system such as those properties that were not in the original test specifications [10]. Moreover, the authors classify an emergent behavior of the system under study as desired, undesired, and not yet specified.

Emergent properties and emergent behavior are well-known phenomena in systems engineering. Designers must pay careful attention to emergent behavior to ensure the correctness of a system, especially in the critical systems domain. A capital example of an emergent undesired and not specified behavior happened in a Nuclear Power Plant. Reviewing this example is indicative of the impact of a system architecture to mitigate these emergent behaviors.

The Three Mile Island accident, known as TMI-2, occurred in a nuclear reactor in 1979 [11]. TMI-2 is one of the most studied accidents for its lessons learned. Although the TMI-2 accident did not cause deaths or contamination cases, it has been a societal shock, with the birth of the antinuclear movement, and modified the standardization process of nuclear safety standards [12]. TMI-2 happened because an accident in the secondary circuit of the nuclear power plant has had consequences on the primary circuit.

TMI-2 Scenario The chain of the main events can be summarized as follows[1]. A simple accident in the secondary circuit automatically involved a safety command.

- First failure: one safety valve was expected to be opened, it was closed after a maintenance procedure. This error indirectly contributed to a reactor overheating.
- Second failure: another safety valve (PORV - pilot-operated relief valve) received the "`close command`" however it remained in the open position.
- Software design error: data reported on the operational control monitor were referred to send command status "`close the valve`" and not on its execution on the system (i.e. actual state of the valve, still opened).

In this situation, the emergent behavior arose from a set of events, never even imagined, which have resulted in inconsistent data that have been presented to the operator. And on this information, the operators made choices, which turned out to be incorrect. The design of the TMI-2 Nuclear Power Plant was robust enough and resilient to mitigate the consequences of this unexpected emergent behavior.

Although TMI-2 demonstrates far less autonomy that modern AI-based systems, it allows us to propose a characterisation for the root cause of some emergent behaviors. Indeed, TMI-2 illustrates that emergent behaviors arise from the conjunctions of multiple minor events whose confluence creates a major safety hazard.

To better understand these conjunctions, one must understand the organizational structure of a system: its architecture. In [13], the author provides a first characterization of an autonomous system from an architectural perspective. An autonomous system is organized around five blocks: Perception, Refection, Goal management, Planning and Self-adaptation. These blocks are organized to fulfill a particular mission and may be subject to emergent behaviors.

[1] See Chapter 9 of [11] for a complete description and analysis of this incident.

The attention to emergent behaviors requires an even more exacerbated analysis when we expand the study from traditional cyber-physical systems, such as a Nuclear Power Plant [14], to autonomous and mobile AI-driven cyber-physical systems. In those later systems, complexity is given by two more components: the supporting runtime architecture made of sensors, processors, and actuators to support the system functions as a decentralized distributed system; and the system's ability to automatically adapt itself to the environment and signals.

Let us focus on the origin of emergent behaviors in IA-CPS. We mentioned we focus on the confluence of minor events that may have a significant safety impact. An IA-CPS covers multiple domains: control, energy, real-time, vision. First, we segregate them by the two high-level ones: CPS and AI. This leads to three potential sources of emergent behaviors, two of which are well-studied:

- emergence in CPS: elements of a CPS architecture may present emergent behavior due to the inherent nature of the interactions between computational (cyber) and physical part, which is surveyed in [15].
- emergence in AI: Similarly, emergence in AI systems is a large topic, under heavy research investigation, e.g. in [16].
- emergence at the AI/CPS boundary: to the best of the authors' knowledge, this situation is less investigated. However, it is also the source of many emergent behaviors. In particular, typical CPS components may present a threat to AI ones, and vice versa. We review such scenarios in the next section.

We note that each domain developed specific engineering methodologies and tailored safety assessment processes. Several groups analyze emergent behaviors within CPS or AI systems. We claim this line of research must be complemented by research work on the AI/CPS boundary.

4. Emergent behaviors at the AI/CPS boundary

Artificial Intelligence and CPS are two disjoint research and engineering communities, yet their collaboration. We mentioned in the previous section that the coupling between AI and CPS topics be the source of emergent behaviors. We motivate the existence of these emergent behaviors by introducing one support case study (in figure 1) and multiple scenarios.

The robot in figure 1 has a LEN (Lifelong Exploratory Navigation) architecture [17]. LEN is based on a cross-layer architecture from the robot's sensors to the occupation grid map, to the generalized Voronoi graph (GVG) until to the navigation and exploration algorithms and rising up to the high-level AI, and from here down across the architecture until the actuators. LEN allows robot navigation and space exploration in dynamic and unknown environments.

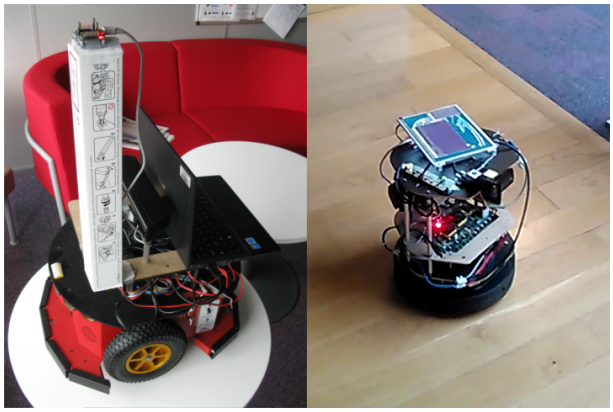

Figure 1: The autonomous mobile robot having AI

LEN is developed by CEA. We chose this example, because for some years now, we have seen an industrial trend to the development of mobile robots, with different levels of autonomy with/without AI, in different civil applications, such as household robots or toys. Compared to the past, where robots were mainly used in industry with little direct contact with the operator, the current type of applications target the masses and have a closer interaction to humans. This trend is expected to grow over the next decades.

To increase the presence of robots near humans, a few guarantees should be met. First, robots have to operate safely and not hurt users. Second, the user should not be required to program, reset or maintain the robot. Third, unlike a factory, the environments in which the robot operates are not controlled in any way by the constructor or the developer. Thus, the robot must adapt itself to its environment, manage its own resources (memory, computing power and battery), and perform its missions.

4.1. Resource Management

Embedded or Edge AI is resource-demanding [18]. In the context of robotic IA-CPS, these resources are (but not limited to): (1) the use of energy (battery), (2) the extensive use of the processor, and (3) the use of the network to exchange messages with other systems.

Let us be consider the following mission for the LEN robot: to bring an object from location A to location B, without having a predefined cartography of the environment in memory. Moreover, the robot has to recognize a given person to whom to deliver the transported object.

To move from A to B, the robot requires a navigation and exploration system. Such a subsystem may require an excessive use of the processor and then an important amount of energy for its execution.

Two emergent behaviors may arise from a bad coupling between AI functions and the supporting CPS platform. In this case, we show that the CPS architecture may have an adversarial effect on the AI functions.

First, the aforementioned subsystem may use so many resources that it interferes with the nominal mission of the robot. For example, the subsystem requires and consumes the entire battery, making impossible to fulfill the mission. This potential conflict situation increases with the deployment of autonomous subsystems in the robot (e.g., the perception of the environment, the recognition of the person, ensuring the security level of the document). An on-line optimization (i.e., after the development of subsystems from different teams and their deployment in the robot) could involve emergent behaviors. Second, the AI or robotics function may induce a significant computation overload. It is well-known that the performance evaluation of robotics platform is a challenge, e.g. [19]. Uncontrolled CPU workload may trigger timing violations, leading ultimately to errors in computations that could impact multiple subsystems.

An approach to overcome the emergent behaviour by guarantying performance of IA-CPS could be twofold. From one hand, the design and development of lightweight solutions for autonomous subsystems. For example, LEN uses resources efficiently [20]. From the other hand, to understand which solutions can be developed to detect, control and mitigate emergent behaviors, by guaranteeing the performance of the AI embedded in the system. LEN is built on a cross-layer architecture. This latter includes two macro blocks. Each block has more layers. The low-level macro-block (the one from the sensors to GVG) implements the safety-related control-command and is devoid of AI functionality. The high-level macro block contains Machine Learning-based making-decision and interfaces with other AI-driven functionality of the robot. The management of errors and emergent behaviors between layers uses a contract-based approach [17] (See section 5.2).

4.2. Cybersecurity and Cyber-Physical Security

We mentioned that IA-CPS are ultimately networked software-intensive systems. As such, they can be subject to cyber-security attacks, i.e. the malicious tampering of some CPU or software functions. In addition, sensors and actuators can be subject to additional attacks, leading to cyber-physical attacks [21]. In these situations, the system may no longer fulfill this mission: data are corrupted, the system reprogrammed, or resource mismanaged, etc. A system that has been compromised exhibit an inconsistent behavior, akin to an emergent behavior.

In [22], we showed that from a system perspective, a faulty sensor and an attacked sensor have a similar behavior. We propose a review of attacks and faults and how to detect them. In particular, we show that the architecture of an IA-CPS can be extended with specific fault or attack detectors to improve the system resilience.

Furthermore, adding a fault or attack detector may in turn generate new emergent behaviors in the case of false positive: the system may overreact to non-existent attacks. This is a well-known paradox due to fault detectors that are not absolute, but rely on some state evaluation.

4.3. Conclusive Remarks

In this section, we listed some scenarios for emergent behaviors at the boundary of AI and CPS. We focused on non-functional properties and discussed security and performance (energy and CPU). This could be extended to all timing aspects (latency, jitter, scheduling, ...), but also safety.

We note that the increased complexity of those systems calls for an impossible holistic engineering approach: one needs to “tame" emergent behaviors that stem from AI, and CPS subdomains, but also evaluate the cross-domain impact of CPS non-functional properties to AI and vice-versa.

We deem this approach as impossible not from a scientific perspective, but from an economic one: we used TMI-2 as an example of a system that mitigated a significant emergent behavior. However, the nuclear industry can spend more time and effort to improve its design. This is not possible for general CPS: the time to deliver a new product should be reduced.

In this context, we consider one should instead focus on assuring the system is sufficiently resilient rather than safe.

5. From Safety assessment to Resilience assurance

Of the extensive literature on methodology and analyses for safety-related properties, we here discuss existing work that is closer to our approach. This literature ranges from traditional analysis techniques, such as FTA and FMEA, to pattern-based, contract-based, GSN (Goal Structuring Notation).

An interesting and mature approach is the one based on the use of design patters for achieving security or safety objectives [23] and for which today we have encouraging and positive feedback. In this context, design patterns provide an architectural description that improve the resilience of the system to specific scenarios,

combined with fragments of an assurance case. As an extension, the authors in [24] promote a methodology that combines patterns with contract-based and GSN - thus obtaining a modular and structural result for safety argument.

The main benefit of patterns and, more generally, of techniques such as GSN and contracts, is the ability to ensure both safety- and cybersecurity-related properties [25, 26]. In [27], the authors propose a safety and security co-engineering framework, based on patters of process and argumentation.

However, despite the essential help of the above approaches, safety assurance and safety analysis remain very expensive. They require redundancy, diversity and independence at software, architectural and physical levels. Such an option is not always possible in IA-CPS , where the limitations of the physical space and cost constitute stronger constraints for IA-CPS applications than traditional ones. In many cases, heterogeneous architectures cannot be applied.

To overcome these difficulties, an interesting approach, albeit not without live debates in the community (see e.g. [28]), is to widen the safety definition to Safety I and Safety II [29]. In [30], the author states that "a system cannot be resilient, but a system can have a potential for resilient performance". Hollnagel proposes to change the classical safety analysis process that focuses mainly on reducing the number of adverse outcomes by taking into account the success stories that tend to become invisible and insignificant, because they are considered as normal, i.e. as planned. Hollnagel introduces the following definitions [29]:

Safety-I aim is to be sure that the number of unwanted outputs will be as low as possible.

Safety-II concerns the condition of being certain that the success of outputs will be as high as possible.

In [28], the author introduces the notion of Safety III as follows

Safety-III freedom from unacceptable losses as identified by the system stakeholders. The goal is to eliminate, mitigate, or control hazards, which are the states that can lead to these losses.

Moreover, Lavenson argues the importance of improving and extending Safety-III. We introduce resilience as a subclass of Safety III. Resilience could allow us to release some hard constraints related to safety (e.g. redundancy, sensor quality) and provide a given level of no longer safety but system resilience. In this regard, we should distinguish between Exogenous and Endogenous resilience.

Endogenous Resilience is the ability of the system to detect and manage internal faults. For example, an autonomous mobile robot should be resilient to internal software errors, manage battery properly, execute a command, detect physical damage of sensors and control it, etc.

Exogenous Resilience deals with the system's external environment in which it is operated., e.g., avoiding an obstacle or malicious attacks

To ensure Safety-I, traditional techniques (such as redundancy) are difficult to deploy on IA-CPS systems where the physical space of the product and its final cost are much more limited than traditional systems. Techniques on resilience appear more appropriated for this new generation of systems and could be based on the design and the control of a modular architecture, having different levels of trust.

5.1. Conclusive Remarks

Unlike traditional critical systems, in IA-CPS systems, the emergent behavior is expected to increase. It could arise from how IA-CPS are used and from the interaction of IA-CPS with its environment. In [1], for example, the authors discuss how "the assurance is insufficient to address the emergent properties derived from the network weights" with respect to the avionics safety norms DO-178C and DO-254. Therefore, preventing accidents in IA-CPS requires using models that include the entire socio-technical aspects and treat safety as a dynamic control problem. Future intelligent autonomous systems need to be able to appraise safety issues in their environment, self-learn from experience and interactions with humans, and adapt and regulate their behavior.

5.2. Way Forward

Within the IC and Digital System Division, at CEA LIST, we are interested in Trustworthy Artificially Intelligent Adaptive Autonomous CPS. As discussed in the previous sections, these systems can be affected by emergent behaviors, particularly if we consider them in dynamic and unpredictable environments. In this research context, we focus on the navigation and exploration system and we use LEN [17] as an excellent case study to experiment the achieved scientific results. More precisely, we would like firstly to understand what methods and techniques are needed to guarantee and ensure a sustainable and trustable low-level architecture, i.e from sensors to the occupancy grid and the Generalized Voronoi Graph (GVG) and then back down to the actuators. In this first architectural block, the main safety control-commands are implemented (including the control of the admissible incertitude). As argued in section 4.1, the design of lightweight solutions has a paramount importance to reduce the risk of uncontrolled emergent behaviours. In

LEN, we wish to (1) individualize and categorize the other factors that could lead to emergent behaviors; (2) qualitatively and quantitatively assess resilience for LEN, by adapting approaches such as [31].

At the SEI, the MBE team is working on the definition of languages and tool-supported processes to engineer safety-critical systems. This encompasses model-based techniques such as the AADL architecture description language along with code generation techniques, safety analysis capabilities. We have developed a collection of techniques to ensure the correctness of code generated from models, along with model checking or Digital Twins capabilities [32]. These provide the foundations to fully analyze a system either analytically or through simulations, with a close link to the engineering models. The SEI is currently engaged in a project to further strengthen the link between MBSE, resilience in the context of IA-CPS. Most notable, we plan to address solutions to all challenges highlighted in the previous section.

6. Conclusion

In this paper, we have discussed emergent properties. We have shown through the TMI-2 nuclear accident, how undesirable emergent behavior can occur in traditional systems. In Increasingly-Autonomous Cyber-Physical Systems, emergent properties are becoming more critical. Unlike traditional application domains, the potential consequences of undesired emergent behavior may be more difficult to mitigate, given the limitation of the physical space and cost. We discussed the emergent properties from three perspective, performance of AI, cyber-security and resilience, and we have briefly illustrated some of the solutions available in the literature. From our analysis of the state of the art and practice, we advocate that the resilience assurance of Increasingly-Autonomous Cyber-Physical Systems requires a careful review of the coupling between AI functions and the architecture of the CPS that hosts it.

Acknowledgments

We thank Laurent Soulier (CEA) for the discussions on LEN.

References

[1] E. E. Alves, B. Devesh, B. Hall, K. Driscoll, A. Murugesan, J. Rushby, Considerations in Assuring Safety of Increasingly Autonomous Systems, Technical Report NASA/CR-2018-220080, NF1676L-30426, 2018.

[2] National Transport Safety Board, Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian Tempe, Arizona March 18, 2018, Accident Report NTSB/HAR-19/03 PB2019-101402, National Transport Safety Board, 2019.

[3] P. Koopman, B. Kuipers, W. H. Widen, M. Wolf, Ethics, safety, and autonomous vehicles, Computer 54 (2021) 28–37. doi:10.1109/MC.2021.3108035.

[4] A. Vogelsang, M. Borg, Requirements engineering for machine learning: Perspectives from data scientists, CoRR abs/1908.04674 (2019). arXiv:1908.04674.

[5] A. Kane, O. Chowdhury, A. Datta, P. Koopman, A case study on runtime monitoring of an autonomous research vehicle (ARV) system, in: Runtime Verification International Conference, volume 9333 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 102–117. doi:10.1007/978-3-319-23820-3_7.

[6] L. Myllyaho, M. Raatikainen, T. Männistö, T. Mikkonen, J. K. Nurminen, Systematic literature review of validation methods for ai systems, Journal of Systems and Software 181 (2021) 111050. doi:https://doi.org/10.1016/j.jss.2021.111050.

[7] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, B. Zhou, Trustworthy AI: from principles to practices, CoRR abs/2110.01167 (2021). arXiv:2110.01167.

[8] N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, P. Tonella, Taxonomy of Real Faults in Deep Learning Systems, arXiv:1910.11015 [cs] (2019). ArXiv: 1910.11015.

[9] A. Nikanjam, M. M. Morovati, F. Khomh, H. B. Braiek, Faults in Deep Reinforcement Learning Programs: A Taxonomy and A Detection Approach, arXiv:2101.00135 [cs] (2021). ArXiv: 2101.00135.

[10] J. Sifakis, D. Harel, A. Marron, Autonomics: In search of a foundation for next-generation autonomous systems, Proceedings of the National Academy of Sciences of the United States of America 117 (2020) 17491 – 17498.

[11] J. Couturier, M. Schwarz, Current State of Research on Pressurized Water Reactor Safety, Science and Technology Series, EDP Sciences, 2018.

[12] J. Samuel Walker, Three Mile Island: A Nuclear Crisis in Historical Prospective, 2004.

[13] J. Sifakis, Autonomous Systems – An Architectural Characterization, Springer International Publishing, Cham, 2019, pp. 388–410. doi:10.1007/978-3-030-21485-2_21.

[14] M. D. Franusich, Security Hardened Cyber Components for Nuclear Power Plants, Technical Report Grant No. DE-SC0013808, US Department of Energy, Office of Science, Chicago Office, 2016.

[15] S. Tyszberowicz, D. Faitelson, Emergence in cyber-physical systems: potential and risk, Frontiers of Information Technology & Electronic Engineering 21 (2020) 1554–1566. doi:10.1631/FITEE.2000279.

[16] Z. Li, C. Sim, M. Hean Low, A Survey of Emergent Behavior and Its Impacts in Agent-based Systems, in: 2006 IEEE International Conference on Industrial Informatics, IEEE, Singapore, 2006, pp. 1295–1300. URL: http://ieeexplore.ieee.org/document/4053581/. doi:10.1109/INDIN.2006.275846.

[17] F. Mayran de Chamisso, D. Cancila, L. Soulier, R. Passerone, M. Aupetit, Lifelong Exploratory Navigation: an Architecture for Safer Mobile Robots, IEEE Design and Test (2019).

[18] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, J. Zhang, Edge intelligence: Paving the last mile of artificial intelligence with edge computing, Proceedings of the IEEE 107 (2019). doi:10.1109/JPROC.2019.2918951.

[19] T. Kronauer, J. Pohlmann, M. Matthé, T. Smejkal, G. P. Fettweis, Latency overhead of ROS2 for modular time-critical systems, CoRR abs/2101.02074 (2021). URL: https://arxiv.org/abs/2101.02074. arXiv:2101.02074.

[20] F. Mayran de Chamisso, L. Soulier, M. Aupetit, Robust topological skeleton extraction from occupancy grids for mobile robot navigation, in: Proceedings of the twentieth national congress on Shape Recognition and Artificial Intelligence (RFIA'16), 2016.

[21] A. Humayed, J. Lin, F. Li, B. Luo, Cyber-physical systems security—a survey, IEEE Internet of Things Journal 4 (2017) 1802–1831. doi:10.1109/JIOT.2017.2703172.

[22] L. Zhai, A. Kanellopoulos, F. Fotiadis, K. G. Vamvoudakis, J. Hugues, Towards intelligent security for unmanned aerial vehicles: A taxonomy of attacks, faults, and detection mechanisms, in: AIAA SCITECH 2022 Forum, 2022. doi:10.2514/6.2022-0969.

[23] C. Preschern, N. Kajtazovic, A. Höller, C. Kreiner, Pattern-based safety development methods: overview and comparison, in: EuroPLoP '14, 2014.

[24] I. Sljivo, G. J. Uriagereka, S. Puri, B. Gallina, Guiding assurance of architectural design patterns for critical applications, J. Syst. Archit. 110 (2020) 101765.

[25] J. Dobaj, D. Ekert, J. Stolfa, S. Stolfa, G. Macher, R. Messnarz, Cybersecurity threat analysis, risk assessment and design patterns for automotive networked embedded systems: A case study, JUCS - Journal of Universal Computer Science 27 (2021) 830–849.

[26] S. Mouelhi, E. Laarouchi, D. Cancila, H. Chaouchi, Predictive Formal Analysis of Resilience in Cyber-Physical Systems, IEEE Access 7 (2019).

[27] H. Martin, R. Bramberger, C. Schmittner, Z. Ma, T. Gruber, A. Ruiz, G. Macher, Safety and Security Co-engineering and Argumentation Framework, in: Computer Safety, Reliability, and Security, Springer International Publishing, 2017, pp. 286–297.

[28] N. Leveson, Safety III: A Systems Approach to Safety and Resilience, http://sunnyday.mit.edu/safety-3.pdf, 2020.

[29] E. Hollnagel, R. Wears, J. Braithwaite, From Safety-I to Safety-II: A White Paper, 2015.

[30] E. Hollnagel, Rag - the resilience analysis grid, Resilience engineering in practice: a guidebook. Ashgate Publishing Limited, Farnham, Surrey (2011) 275–296.

[31] R. Bloomfield, G. Fletcher, H. Khlaaf, L. Hinde, P. Ryan, Safety Case Templates for Autonomous Systems, CoRR (2021).

[32] J. Hugues, A. Hristosov, J. J. Hudak, J. Yankel, Twinops - devops meets model-based engineering and digital twins for the engineering of cps, in: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS '20, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3417990.3421446.