A Taxonomy for Architecting Safe Autonomous Unmanned Systems

Yixing Luo
Peking University
Beijing, China
yixingluo@pku.edu.cn

Haiyan Zhao
Peking University
Beijing, China
zhhy.sei@pku.edu.cn

Zhi Jin
Peking University
Beijing, China
zhijin@pku.edu.cn

ABSTRACT

Autonomous Unmanned Systems (AUSs) emerge to replace human operators for better efficiency and effectiveness, especially in harsh and dangerous environments which frequently imply uncertainty. Safety has become one of the top concerns for AUS designs. To address AUS safety concerns systematically, we aim to establish a comprehensive taxonomy of AUS safety and provide a safety-by-design framework for architecting safer AUSs. We conduct a systematic literature review on 65 primary studies and analyze them from three perspectives: system and environment features, safety threats, and countermeasures. We adopt feature models to organize the survey results and establish a taxonomy for AUSs safety issues. Based on the taxonomy, we figure out a reference architecture that integrates three control loops dealing with the uncertainty of operating environments, external threats and system deviations, respectively. Our survey reveals that AUS safety is still a formative field and presents a taxonomy for AUSs safety issues and a safe-by-design framework for architecting safer AUSs.

KEYWORDS

Self-adaptive systems, software architecture, feature models, autonomous unmanned systems

ACM Reference Format:
Yixing Luo, Haiyan Zhao, and Zhi Jin. 2022. A Taxonomy for Architecting Safe Autonomous Unmanned Systems. In *13th Asia-Pacific Symposium on Internetware (Internetware 2022), June 11–12, 2022, Hohhot, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3545258.3545290

1 INTRODUCTION

Recent years have witnessed a dramatic growth of Autonomous Unmanned Systems (AUSs) that can function independently without human intervention, e.g., unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs), and unmanned underwater vehicles (UUVs) [25]. However, the wide application of AUSs brings potential safety risks. Actually, numerous accidents have been reported. For example, Google's self-driving cars were involved in 14 accidents from 2010 to 2015 [87]; a Uber's self-driving car killed a pedestrian in 2018 [33]; a Tesla in Autopilot mode crashed into an overturned truck in 2020 [26]. The Washington Post has published that there have been hundreds of "CLASS A" crashes of military drones, and each of them results in at least $2 million loss [63].

This presents a dilemma to AUSs between potential benefits and risks. In general, for AUSs applied to safety-critical scenarios (i.e., transportation, search and rescue, surgery, etc.), safety concerns should involve all stakeholders, as a tiny mistake could result in human injuries, significant property loss, or damage to the environment. In fact, safety has already become an unignorable concern in AUS designs. Responsible authorities or organizations have formulated safety regulations and standards (e.g., CAP 722 for unmanned aircraft systems by Civil Aviation Authority [7], ISO 10218 of safety requirements for industrial robots [40]).

Existing works involve hazard analysis and risk reduction at design time, and the self-adaptation at run time. Hegde et al. [37] propose a Bayesian Belief Network to model the risks affecting autonomous inspection, maintenance, and repair operations and provide decision-support for system developers in the uncertain and dynamic subsea environment. James et al. [54] put forward a regulatory-based integrated approach to system safety and risk analysis of the Unmanned Aircraft Systems. Banda et al. [10] elaborate a systemic hazard analysis and define safety controls for mitigating at the concept design phase of autonomous vessels within their operative context.

At run time, measures for assuring AUSs safety concentrate on the self-adaptation of the system during the close interaction with environment entities like obstacles, other systems, and humans. Zhong et al. [90] propose a hybrid path planning method based on A* algorithm and adaptive window approach for real-time obstacles avoidance of mobile robots in large-scale dynamic environments. Noh [60] proposes a decision-making framework to determine appropriate maneuvers for an autonomous vehicle to navigate an intersection safely and efficiently, even in the face of violation vehicles. Jan et al. [42] use a behavior-based architecture with a Pedestrian Interaction System for the efficient and safe navigation of autonomous shuttle in a pedestrian area.

However, these works by themselves are all case-by-case but not on the methodological level. For designing a safer AUS, it is highly expected to have general guidelines for helping AUS designers to systematically examine the safety issues. Such guidelines should provide recommendations that outline the necessary features for ensuring the safety of AUSs from the perspectives of both the systems and their surrounding environments. It also needs to provide a safety-by-design framework that consists of components in charge of implementing the features.

For developing such a guideline, we conduct a systematic literature review [47] to extract the features of AUS safety concerns and

Internetware 2022, June 11–12, 2022, Hohhot, China

ACM ISBN 978-1-4503-9780-3/22/06…$15.00
https://doi.org/10.1145/3545258.3545290

develop a taxonomy. The review compiles 65 primary studies on safety concerns in AUSs, and the studies are selected from main conferences and journals between 2009 and 2020. All of the studies are investigated from three perspectives: the software system and the environment features, the safety threats, and the countermeasures. The surveyed results are organized in feature models to clarify the attributes of safety concerns shared amongst AUSs.

In terms of the safety issues identified in the taxonomy, we propose and design a reference software architecture for enabling safer AUSs. With the observation that the essential source of the safety concern is uncertainty, this architecture integrates three control loops for dealing with the uncertainty of the operating environments, the external threats, and the system deviations, respectively. It then explicitly embeds the necessary components for including the design recommendations or countermeasures to threats. The main contribution of this paper is to establish a taxonomy for the AUS safety concerns and a reference architectural model for AUS safety design.

2 FEATURE EXTRACTION FOR AUS SAFETY CONCERNS

In this section, we briefly discuss the safety concern for AUSs, set up a top-level taxonomy for features of AUS safety concerns, and present the process of feature extraction.

2.1 Safety Concerns for AUSs

Safety is a top concern to the public regarding AUSs that have no human operators to act as a fail-safe. Leveson considered the safety in the context of software engineering as "the freedom from accidents or losses" [49]. Here, safety is regarded as a property of the software-intensive systems and is concerned with a malfunction in hardware or bugs in the software that could lead to safety hazards. In the automotive industry, safety is well covered by ISO 26262 which includes systematic processes for stipulating how software should be defined, developed and tested to conform to good systems engineering practices [66]. However, for AUSs, it is not enough to merely functioning correctly [48], as during the operation, uncertainties and dynamics in the environment may pose safety threats [19]. With incomplete assumptions of system states and environmental conditions, unanticipated accidents like physical injury of people and loss or damages to equipment/property may occur in the interaction between AUSs and the environment. Take the operating environments for AUSs into account, Avižienis et al. referred the safety as "the absence of catastrophic consequences on the user(s) and the environment" [9]. Luo et al. [53] elicited the safety requirements from an environment-centric view for safe autonomy of AUS as hazard-elimination and conflict-avoidance. In short, to analyze safety concern for AUSs, taking both the system- and environment-centric view has become a consensus. In the following, we summarize the safety concerns extracted from the perspectives of both environment and system and as a result; such that we obtain a taxonomy of AUS safety concern.

2.2 Feature Extraction Process

To extract the safety concerns of AUSs, we conduct a systematic literature review (SLR) [47].

Table 1: Research Questions and Top-level Classification of Features for AUS Safety Concerns.

RQ1:	What are the safety concerns from the perspective of AUSs?
RQ1.1:	**System Component** What components are equipped in AUSs?
RQ1.2:	**Internal Safety Threat** What are the safety threats arise from AUSs?
RQ1.3:	**Countermeasure for Internal Safety Threat** How to address safety threats arise from AUSs?
RQ2:	What are the safety concerns from the perspective of the interaction between AUSs and environments?
RQ2.1:	**Environment Characteristic** Which characteristics do the operating environments hold?
RQ2.2:	**External Safety Threat** What are the safety threats arise from the environments?
RQ2.3:	**Countermeasure for External Safety Threat** How to address the safety threats arise from the environments?

2.2.1 Top-Level Classification of Safety Concerns. The top-level classification of the safety concerns is given in Table 1. First, safety concerns are examined from the perspective of both the system and the environment. As to the safety concerns from the system-centric view, we investigate (1) the system components which are responsible for their autonomous behavioral decision-making; (2) the internal safety threats whose causes lie in the system; (3) the corresponding measures that are designed for tackling the internal safety threats. As to the safety concerns from the environment-centric view, the prerequisite is a systematical investigation of different conditions of the operating environment where safety threats may arise. Then, the external safety threats due to the dynamics and uncertainties of the environment are analyzed, and the countermeasures that can be used to prevent or control these threats are identified. Therefore, all articles are investigated concerning two main research questions (RQ1 and RQ2) from the perspective of the system and operating environment. The two research questions are be divided into three separate parts, which can be turned into sub-questions: the system components or the environment characteristics, the safety threats that arise from the AUS or the environments, and the countermeasures.

2.2.2 Search Process. The review process was conducted as follows. The following prominent indexing services were used during the literature review: IEEE Xplore, ACM Digital Library, Science Direct, and Springer. The selection of these electronic databases is guided by their high accessibility and efficient means to conduct systematic literature reviews in software engineering [14]. We elicit two keywords from our research questions: "safe" and "autonomous system", and construct the search string using Boolean "AND" to join these main terms and "OR" to include synonyms. Studies are retrieved using the search string "safety" (OR "failure" OR "error" OR "fault" OR "hazard" OR "risk" OR "collision") AND "autonomous system" (OR "UAV" OR "drone" OR "vehicle" OR "vessel" OR "robot"). After applying inclusion criteria and exclusion criteria to the initial search results, the review ended with a selection of 65 primary studies from 2009 to 2020, which are analyzed in this research [71].

3 TAXONOMY FOR FEATURES OF AUS SAFETY CONCERNS

This section describes the results of our literature study. As the six top-level features of AUS safety concerns are mapped with the research questions RQ1.1-RQ2.3, the answers to these questions

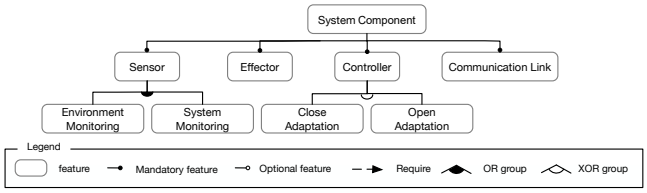

Figure 1: Feature model of system component of AUSs.

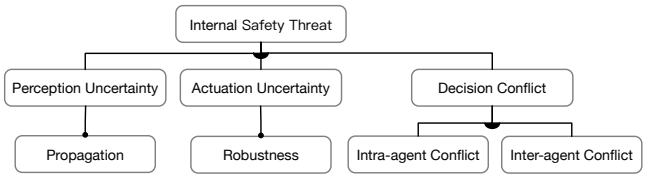

Figure 2: Feature model of internal safety threats.

are represented in six feature models, which together construct an overall taxonomy of features for AUS safety concerns. To summarize the features included to answer each question and how the features are related, we use hierarchical feature models in this paper. Specifically, the relationship between a parent feature and its child features are categorized as *Or* and *Alternative (xor)*. Either child feature could also be a *Mandatory* or *Optional* one as the decomposed of the parent feature. In addition to the parental relationships, the relationship between child features can be cross-tree constraints like *Require*, *Exclude* etc.

3.1 Safety Concerns in System (RQ1)

3.1.1 System Component (RQ1.1). We analyze the features of the AUS from the perspective of their software components. These individual components in the AUS usually follow a pipeline, linking sensor inputs to effector outputs [56]. As the core part of an AUS, the controller is responsible for analyzing the information collected by sensors, planning, and making decisions for the effectors [56] to execute. The feature model describing the components of AUS is shown in Fig. 1.

Sensor. The main responsibility of sensors is to monitor and collect information related to the system states and environmental conditions. External environmental information is collected through environmental monitoring, while internal system information is collected via system monitoring. *(i) Environment Monitoring:* In the surveyed literature, sensors that AUSs equipped with are responsible for environmental monitoring and data analysis. Cameras for obstacle detection and localization are widely deployed in the AUS to monitor changes in the environment to reduce the risk of collision [56]. Sonar and LiDAR are used for distance measurement. However, an inherent defect of sensors is that they are not fully reliable [50]. The data collected can be biased due to sensor noise [4, 80], raw signal noise [29]. *(ii) System Monitoring:* Separately from the environment monitoring, the internal state and system properties of an AUS should also be monitored at runtime. The operational performance is monitored based on the safety indicators in [74] and similar monitoring frameworks for autonomous systems are also proposed in [35, 38, 55]. Problems such as running out of energy or component failures should be instantly detected so as to keep the system running smoothly in abnormal situations.

Controller. As the core of AUS, the controller determines its behaviors at runtime. The adaptation modes of the controller affect the way for the AUS to cope with changes in dynamic environments and the system. For an AUS, the adaption modes of the controller can be either close or open [64]. *(i) Close Adaptation:* A controller in the close adaptation mode has a fixed number of adaptive actions determined at design time, and no new behaviors or alternatives can be introduced during runtime. *(ii) Open Adaptation:* In a controller with the open adaptation mode, self-adaptive actions can be extended at runtime with new alternatives using online learning algorithms, e.g., gaussian process modeling [79], dynamic planning [89]. AUSs with open adaptive strategies gain the advantage in more complicated tasks than those limited to close adaptive strategies.

Effector. The effector is responsible for applying the actions determined by the decision-making process in the controller [64]. For example, the pedals and steering wheels, as the effectors of the autonomous vehicle, adjust the acceleration/deacceleration and direction of the vehicle based on the trajectory decided by the path planner. Some effectors of the AUS may be involved in the physical Human-Robot Interaction (pHRI), and it is important to take action uncertainty into consideration [51] in this scenario.

Communication link. They are responsible for transmitting data and control signals among different components, and the stable communication between the AUS and the cloud, ground control systems, or other systems. For example, for a drone with limited computational and storage resources, computation tasks may be outsourced to the cloud [36], and communication links are established between the drone and the cloud for data transformation. For semi-autonomous unmanned systems, control signals are transformed in the communication links between the system and the ground stations [22]. Attacks to communication links that are responsible for transferring control signals may cause the system to lose control and destroy the environment and system [36].

3.1.2 Internal Safety Threat (RQ1.2). The internal safety threats arise from the vulnerability of the critical components in a self-adaptive AUS we mentioned before. According to the survey studies, internal safety threats mainly are perception uncertainty, actuation uncertainty, and conflicts in decision-making, resulting from the sensor, effector and controller, respectively. The feature model of internal safety threats is shown in Fig. 2.

Perception Uncertainty. Considering the operating environment is not under the control of the AUS, there may be uncertainty in terms of what is sensed by the sensors [84]. In the pipeline workflow of an AUS, the system interacts with the physical environment through sensor inputs and effector outputs. Perception uncertainty arising from sensor inaccuracy could be propagated to the other components and affect the decision-making of the controller. If a sensor neglects obstacles or confuses them with the background due to its poor performance, its inaccurate understanding of the situation may lead to the wrong action being planned by the controller. In addition, erroneous data and states may also propagate between the different task phases of the system [20], and subsystems [54].

Actuation Uncertainty. Due to the dynamics and unpredictable operating environments for AUSs, there also may be uncertainty in

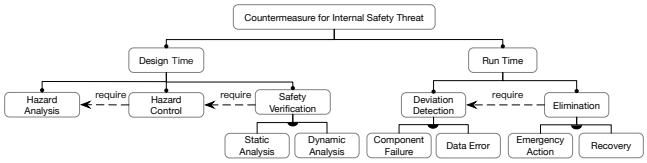

Figure 3: Feature model of countermeasures for internal safety threats.

terms of the outcomes of the effectors [84]. Actions of the AUS can be disturbed by dynamic changes in environmental conditions or component failures that hinder the system from operating correctly. The accumulated actuation uncertainty may affect the robustness of the controller and even cause the AUS to malfunction or be unable to complete its goals.

Decision Conflict. Due to the flawed design of the controller, decision conflicts may arise in the decision-making process. For a single-agent AUS, conflicts may arise when achieving different requirements simultaneously. For a multi-agent AUS consisting of a team of robots, decision conflicts may arise from the competition between team members. *(i) Intra-agent:* For a complex and software-intensive AUS, they could be built consisting of independent units of functionality, e.g., autonomous cars may include independent driving functions like automated emergency braking (AEB), adaptive cruise control (ACC), etc. Suboptimally or wrongly designed controllers or control policies can pose safety threats to the system. Abdessalem et al. [1] discuss the safety issues due to interactions between different units of functionality in self-driving cars when several driving functions control the same actuators to satisfy their own requirements. *(ii) Inter-agent:* In a multi-agent AUS with a distributed decision-making architecture, each agent may be self-interested. Decision-making is likely to depend on local conditions, with limited consideration of global performance and constraints [89]. Therefore, it is challenging to achieve global objectives when the chances of friendly unintentional decision conflicts increase, and all agents (a team member of the AUS) are self-interested and competing for limited resources.

3.1.3 Countermeasure for Internal Safety Threat (RQ1.3). All the countermeasures can be classified into measures carried out at design time or at runtime, as shown in Fig. 3.

Design Time. There is a three-step process for safety assurance of the AUS at design time, i.e., hazard analysis, hazard control, and safety verification. [49].

Hazard Analysis: This process is to discover the root cause of hazards from several aspects. Analysis of the system includes its operation mode [54], system components (e.g., communication link [22], effector [39]), and the relation between different mission phases [6]. Various techniques have been proposed for risk decomposition in a top-down or bottom-up way. Among them, fault tree analysis has been applied to software and hardware risk analysis deductively and starts with identified risks to combine a series of lower-level events and discover the possible causes [85]. Event tree diagrams are an inductive analytical technique that can be applied to complicated systems [57].

Hazard Control: In this process, safety is designed into the system to ensure the appropriate behaviors in response to identified internal safety threats. In terms of sensor noise, one approach is sensor fusion, which combines different inputs of sensors and all available information to overcome the physical limitations of the sensors [36]. The controller design is crucial, and many solutions that can be integrated into the decision-making process in the controller have been proposed to reduce or eliminate hazards [59, 67, 86]. For secure communication channels design, a risk-based multi-criteria decision-making approach for image transfer between UAV and ground control station is proposed by Dursun et al. [22]. Action uncertainty which affects the robustness of the system, is tackled through abstracting the stochastic system transition in the process of motion planning [51].

Safety Verification: Ways to verify the critical safety attributes of the AUS can be classified into dynamic and static analysis to verify whether solutions are proposed to tackle the safety threats. *(i) Static Analysis:* It evaluates the system without running it. A method to generate barrier certificate functions for safety verification is proposed in [75]. Formal methods like model checking can also be useful for providing evidence to regulatory authorities on AUSs safety [82]. *(ii) Dynamic Analysis:* Sometimes, model checking alone is not enough to evaluate the safety of AUS for practical use, as the formal specification may not reflect the actual situation of the user needs (and/or the system and environment), and the proof may contain errors and make incorrect assumptions [68]. Additionally, it is not a cost-effective process as the system size increases, the cost of formal verification rises disproportionately as the scale of system states increases when compared with dynamic analysis, e.g., pivotal variables and functions testing [22], boundary constraints testing [51, 62].

Runtime. Countermeasures for internal safety to be conducted at runtime are often based on a reactive mechanism, in which parameter or property changes work as a trigger for actions to eliminate safety threats. As the prerequisite threats elimination, the monitoring and detecting mechanisms are of greater importance. Then, the system takes reactive policies like recovery and emergency actions, following predefined rules to minimize losses resulting from an unsafe system state.

Detection: The system monitors each internal component or subsystem at runtime to determine whether an accident has occurred based on the hazard list and the event specification. If component failure or data error develops, detailed information about the incident, such as severity, location, or timing, is required for analysis. Monitoring-detection mechanisms allow for online detection of component failure or data error. *(i) Component Failure:* In the surveyed literature, component failure detection is often realized through real-time monitoring and checking whether the safety parameters satisfy their thresholds. Models of the different failure modes [32, 39] and the predefined safety rules [35] reduce the time required to discover safety-critical deviations and to trigger restorative actions. *(ii) Data Error:* To detect the erroneous in sensory data, existing methods are based on signal features (e.g., a comparison between the actual and modulated signals [23]), statistical differences in data (e.g., assuming that the system works with data that follows a normal distribution [34]), and auxiliary equipment (e.g., use the IMU and camera to verify the correctness of GNSS signals [36]).

Elimination: Handling safety threats online, a safety control mechanism is designed in response to any erroneous or undesired event. Widely used methods in this domain are emergency actions

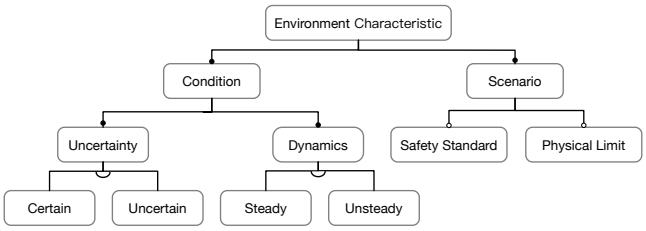

Figure 4: Feature model of Environment Characteristic.

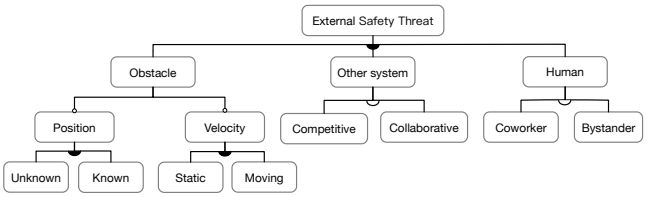

Figure 5: Feature model of external safety threats.

and recovery, which reduce the likelihood of mishaps or recover the system from an unsafe state. *(i) Emergency Action:* AUSs usually have backup plans for emergencies. By recognizing the high possibility of an accident, the system can report and seek confirmation from its user [65], or can autonomously return [36]. Often, emergency actions can be classified into emergency pause (E-pause), which stops system motions temporally, and emergency stop (E-stop), which disables its power module [46]. *(ii) Recovery:* The system recovers from erroneous states by eliminating risk. Data recovery mechanisms are achieved in the AUS by signal reconstruction based on signal features [23].

3.2 Safety Concerns during Environmental Interaction (RQ2)

3.2.1 Environment Characteristic (RQ2.1). There can be different ways to describe the characteristics of the operating environment of the AUSs either with environmental conditions or concrete scenarios, as shown in Fig. 4.

Environmental condition. The characteristics of environmental conditions can be described from their uncertainty and dynamics, i.e., whether there is incomplete or unknown information about the environment and whether the entities in the environment are static or moving.

Uncertainty: If there is noise in the environment, e.g., disturbance from weather [18, 24], unforeseen obstacles or intruders [12], the environmental situation perception may be imperfect or the environment is only partially observed. Obviously, the more uncertain the environment is, the harder it is to perceive.

Dynamics: A steady and stable environment usually has determined parameters and state transitions, while unstable ones cannot be represented by simple linear/nonlinear functions. AUSs that work in a predetermined working space, usually have a predictable environment with potentially unchanging safety threats over time. However, for AUSs which are supposed to operate in a highly open and dynamic environment, the incomplete assumption of the environment may hinder the safe operations like rovers on Mars [79]. Through a combination of those different environmental conditions, the operating environment can be classified into four types, i.e., certain and steady, uncertain and steady, certain and unsteady, uncertain and unsteady.

Scenario. AUSs are designed to operate in numerous concrete scenarios including ground, air, undersea, and sea surface [83]. The most common working space for AUSs are sea, airspace, land which can be further divided into sub-ones. The airspace can be further classified into low altitude for Unmanned Aerial Vehicles (UAVs) working in urban scenarios (below 400 feet [8]) and high altitude for Unmanned Aerial Systems (UASs). In the land scenario, AUSs can be classified into autonomous vehicles for outside use, or industrial/domestic robots used indoors. The scenario of sea area is classified into the above sea for autonomous surface vehicles (ASVs) and undersea for unmanned underwater vehicles (UUVs). Here we refer to robots that are used for manufacturing, surgery, housework, etc. Due to the diversity of implementation of AUSs in concrete scenarios, the environment characterized is also affected by standards and physical limits constrained by the scenario.

Standard: AUSs that operate in some safety-critical scenes must obey applicable laws and industry standards. Standards are ISO-10218 which specifies safety requirements for industrial robots, and IEC-60601 which concludes a series of technical standards for the safety of medical electrical equipment, etc. Autonomous vehicles traveling in urban areas without guidance from human drivers should adhere to driving practices and obey traffic regulations [17].

Physical Limit: For AUSs deployed in concrete scenarios, they are also constrained by the corresponding physical limits, e.g., working space, communication bandwidth, etc. Compared with a UGV operating indoors, the safety of drones in 3-D open spaces such as airspace or sea areas is better to be achieved considering less cluttered environments with obstacles. However, the hydrodynamics of water and wind speed should be taken into consideration for the safe operation of UUVs and UAVs in such environments.

3.2.2 External Safety Threat (RQ2.2). External safety threats result from different environmental conditions, while these conditions are affected by the environmental entities the AUS interacts with [48]. According to the paper [53], external safety threats arise from three main environmental entities, i.e., obstacles, other systems, and human beings.

Obstacle. Due to close interaction with the environment, AUSs are destined to face uncertainties during their operation [88], especially obstacles. The obstacles can be described from whether their positions are known and whether they are static or moving. Through a combination of these features, obstacles can be of four types, i.e., known and static, unknown and static, known and moving, unknown and moving.

Position: Whether being aware of the exact positions of obstacles affects the time cost for making decisions. Prior knowledge of the obstacle accelerates the process of obstacle identification and avoidance. Whereas, unforeseen objects [78] challenge AUS's real-time perception and planning abilities, and increase the risk of collision. For example, possible traffic participants might exist beyond the sensing horizon of autonomous vehicles [59].

Velocity: The movement of obstacles also affects AUS ability to plan a collision-avoidance route. Through sensory perception and recognition algorithms, avoidance of static obstacles can often be

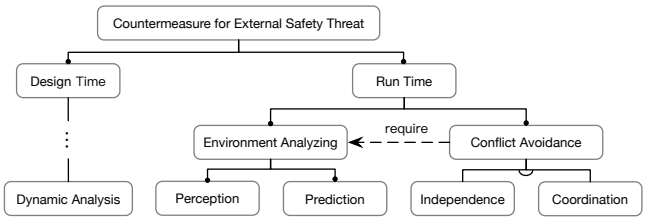

Figure 6: Feature model of countermeasures for external safety threats.

achieved effectively [12, 69, 79]. However, the difficulty in perceiving and predicting the motions of the moving obstacles may hinder the generation of safe behaviors for AUSs.

Other systems. When AUSs are not working alone, they may share resources supplied by their context or rely on information offered by other systems. Therefore, the relationship between AUSs and other related systems can be divided into two categories: competitive and collaborative.

Competitive: Resources are shared among different systems deployed in the same context, e.g., a UAV shares airspace with other aircraft [18, 72, 77], a self-driving car may run in a city alongside other traffic participants [56]. Competitions exist when shared resources or energy are limited. Thus, it is crucial to take into account the capabilities of perception and avoid intruders in the same context [5], especially when designing navigation methods for crowded and cluttered environments (e.g., roads, factories, urban airspace).

Collaborative: Apart from obtaining environmental information from their own sensors, some AUSs may interact with other systems for data sharing. For example, satellites provide GPS signals for autonomous navigation outside [36]. AUSs may be trapped in an erroneous situation if failure or error occurs in the other system AUSs collaborate with, e.g., loss of GPS causes a UAV failing to localize and losing control.

Humans. During the interaction between humans and AUSs, especially in physical Human-Robot Interaction (pHRI), it is essential to ensure human safety. The relationship of AUSs to humans can be divided into two categories: co-workers and bystanders.

Coworker: Collaborative robots are designed to enhance the safety of their users or partners and they are potential to be applied in industry [21]. A crucial aspect for these robots is to maintain a safe workspace for human operators in the human-robot collaboration [76]. Specifically, to guarantee safety during pHRI, the amount of power, energy, or force transferred to a human must be monitored and limited [3]. The risk of human injury can also be reduced through modeling the movement of humans in pHRI [91].

Bystander: In other scenarios, humans are viewed as bystanders or participants sharing the same context with the AUSs. AUS is responsible for the detection and conflict-avoidance with humans (often treated as moving obstacles in practice [42]). For example, it is important for self-driving cars to consider possible traffic participants that might exist beyond their sensing horizon, and to adapt their driving behaviors accordingly [59].

3.2.3 Countermeasure for External Safety Threat (RQ2.3). Countermeasures for external safety threats are summarized in the feature model as shown in Fig. 6, while the runtime solutions are different from those for internal safety threats shown in Fig. 3.

Design Time. For external safety threats, the safety management process can be launched at design time, in which hazard analysis is launched to investigate the risks from the operating environments [57, 62], as well as safety verification [12, 58, 76]. Due to the dynamics and uncertainty of entities in the environment, most efforts are put into the verification and validation of the safety of AUSs in complicated and challenging environment situations through testing which can be performed either on simulated platforms or in real environments.

Testing in the actual environment with physical equipment (hardware-in-the-loop) offers the possibility of detecting problems that may not be found in simulations (software-in-the-loop). In the case of conventional vehicles, manufacturers have a plethora of experiences, i.e., millions of test kilometers making it possible to correct problems and the requirements are clear, the vehicle must be technically fit for the road, complying with all relevant regulations [11]. However, testing in the actual environment can be dangerous and cause serious harm to people or property, and it is also time-consuming and economically expensive. Since we cannot enumerate all the possible traffic scenarios even in real-world testing, scenario-based testing which abstracts features from the real-world scenarios attracts more and more attention. Methods to specify the testing scenarios domain are either data-driven [27] or manually specification [1].

Runtime. Proactive countermeasures are usually taken by AUSs at runtime to prevent accidents and losses such as human injury, property damage, or ecological harm.

Based on assumptions of the state transitions of the environment, these countermeasures are taken to control the occurrence of safety threats and take precautionary measures, instead of adapting after an accident occurs.

Environment Analyzing: The goal of environment analyzing is the perception of environment entities like obstacles, other systems, and humans, and the prediction of their future motions. *(i) Perception:* Information collected from the environment to perceive the environment situation, estimate the semantics of the surrounding scene, and understand its geometry [56]. A Real-Time Location System is a critical component of many mobile agents for autonomous navigation in different environments [34]. For self-driving cars, a semantic scene understanding is to classify the class and state of the surroundings [56, 59].

(ii) Prediction: Even though environment situation perception can identify locations of surrounding objects, many entities are dynamic such as pedestrians and moving obstacles. Through the prediction of motions of these entities, AUSs can avoid dangerous situations, such as a pedestrian possibly stepping onto the road [59]. Given a model of external agent behavior, the motions of these agents can be anticipated [30]. In the absence of a model, inverse reinforcement learning (IRL) can be used to infer the latent reward function for other agents. The inferred reward function can be used to predict external agent motions [91]. A method to learn a conservative model of the world in which actions are guaranteed to be applicable is also proposed [70].

Conflict Avoidance: With the prediction of the movement of people and obstacles around, a safe behavior plan should be found to avoid potential conflicts. For AUSs, there are two types of mechanisms to avoid potential conflicts.

(i) Independence: Decisions are made by the AUS itself. The process of conflict avoidance can be modeled as a Markov decision process (MDP) [72] or a partially observable Markov decision process (POMDP) when the uncertainty of sensor inputs is considered [80]. Safety constrained MDP has also been used in scenarios such as AUSs competing for limited resources [89], or exploration of a partially-known environment with safety constraints [79]. There are several proposed methods to ensure safe motion planning of AUS in a cluttered environment (e.g., forests), including A* and its variants [90], sampling methods of rapidly-exploring random trees (RRTs) [15], and model predictive control [81]. When faced with moving obstacles, like autonomous vehicles at an intersection, appropriate maneuvers are determined based on the prediction of the future paths of observed vehicles [60, 81]. Whereas, in more complicated scenarios, safe motion planning of AUSs must be tackled as a *joint problem* [59] that needs to simultaneously consider various constraints and limitations like observation constraints, dynamic constraints, and computation constraints. However, planning methods based on optimization may incur high time costs and end-to-end approaches are proposed by learning a policy to map from sensor data to control signals using machine learning (ML) techniques [61]. To address the interpretability of the ML models, continuous safe learning based on constraints [52] and imitation learning which combines ML model and feedback control [2] are proposed in the domain of safe autonomous driving.

(ii) Coordination: Decisions are made through negotiation and coordination, and information needed for mission scheduling and collision-avoidance is obtained by communication with people [31, 65] or other systems. Communication protocols can be deployed both intra-AUS or inter-AUS for the system to achieve consensus. Inside the system, communication protocols are used for information sharing [78], while they are also utilized for the coordination between systems, e.g., route scheduling via Traffic Collision Avoidance Systems (TCASs) and aircraft crash avoidance via ADS-B.

4 A REFERENCE ARCHITECTURE FOR AUS SAFETY

Based on the taxonomy (Table 1) we established as the guideline for tackling AUS safety concerns, we propose a conceptual reference architecture for AUSs safety designs as shown in Fig. 7. This architecture has a two-layer control loop. The top layer structures the components of the decision-making mechanism of an AUS in response to safety threats. Considering the safety concerns during the close interaction between AUSs and their operating environments, we also design an interaction loop in the bottom layer for external safety threats detection and elimination. The implementation of the safe architecture for the AUS can be specifically tailored for different applications of AUS through feature selection according to feature models in Section 3. By further integrating the safety concerns into the architectural design of the AUS, and highlighting the environmental properties, we represent the context diagram of the system using the notations of the Problem Frames [41]: rectangular nodes denote the domain entities and edges for their shared interfaces, where arrows highlight the direction of control.

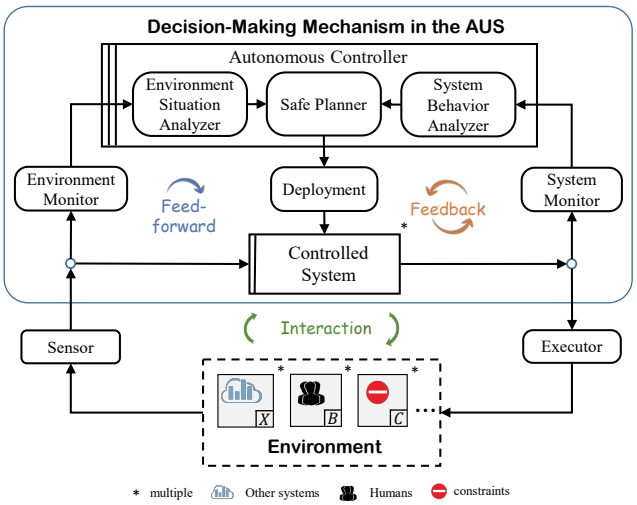

Figure 7: A reference architecture for the safe AUS.

4.1 System Internal Control Loops Design

Based on control theory, an AUS consists of the `Controlled System` and the `Autonomous Controller` [16]. The top layer combines the feed-forward and feedback control loops, and each control loop is refined as a MAPE control loop to handle safety threats through the monitoring, analyzing, planning, and executing components. Safety concerns are modeled as constraints ($\{\phi_1, ..., \phi_m\}$) to guarantee, and they are managed in the `Safe Planner` of the `Autonomous Controller`. Besides safety concerns, the `Safe Planner` module also handles functional requirements which can be instantiated as tasks ($\{T_1, ..., T_n\}$) for the AUS to complete.

The feed-forward control loops is composed of a pipeline of individual components to handle external safety threats proactively. It is the `Environment Monitor` deal with environmental data collected by the `Sensor` to obtain observations about the current surrounding situations. The `Environment Situation Analyzer` is responsible for the proactive checking of whether there would be conflicts with environment entities during the operation. Based on the prediction of `Environment Situation Analyzer`, the `Safe Planner` will treat the *conflict-avoidance* safety concern as a dynamic control problem, as suggested by Leveson [48]. Therefore, an optimal plan for the completion of functional requirements under the guarantee of safety requirements is generated by `Safe Planner`. The plan is further transformed into commands in the `Deployment` module for the `Controlled System` and then translated as control signals to be delivered to the `Executor`.

The feedback control loop is designed to handle internal safety threats reactively. It is the `System Monitor` is designed for monitoring system parameters along with the `Executor` actions. Following the safety concerns of *deviation detection* and *elimination*, system abnormal states, including disturbance of system parameters, failures of certain components or subsystems of the system, and action uncertainty can be detected by the `System Behavior Analyzer`. Once internal safety threats are detected, an alternative plan is expected to be generated by the `Safe Planner` to help the system recover from failures and return to its normal state, even at a reduced level of functional requirements achievement

for fault-tolerance. Otherwise, when the system could not return to its normal state, emergency operations are decided by the `Safe Planner` and delivered to the `Controlled System` for fail-safe.

4.2 Environmental Interaction Loop Design

The bottom layer shows the close interaction between the AUS and its operating environment. The `Sensor` and `Executor` are connected directly with the environment entities for collecting the environmental information and taking actions. With the guidance of an environment modeling-based requirements engineering principle [43], we explicitly highlight the environment entities in the interaction loop. As summarized in Section 3.2, the environment consists of multiple types of domain entities, including other (autonomous) systems sharing the same environment, people around, and physical/logical entities that may pose constraints such as obstacles to avoid, rules/standards to obey, etc. Amongst them, humans are *biddable domains* marked by "B", the physical entities are *causal domains* marked by "C", and the constraints like obstacles are *lexical domains* marked by "X". Differences of these environmental entities in varying degrees of dynamics and uncertainties could have a significant impact on AUS safety.

In safety-critical situations with little tolerance for accidents and losses such as human injuries, property damages, and ecological harm, mitigation of risks is not always sufficient in safety-critical situations. Guided by the safety concerns of *environment-analyzing* and *conflict-avoidance* at runtime, which prevents the system from conflicts, we design a feed-forward control loop and an interaction loop with the environment in the AUS safe architecture. The environment information is captured by the `Sensor`, and then processed by `Environment Monitor`. The `Environment Situation Analyzer` is responsible for environmental perception and predicting whether there would be collisions or intersections with obstacles, other systems, or humans. For a multi-agent AUS, such an analyzer also prediction the motions of other team members. Once there is assumed to be a conflict, the `Safe Planner` will generate a safe action plan for AUS to achieve tasks.

4.3 Comparison with Existing Architectures

We summarize the comparison of existing architectural models for the safety of the AUS as follows.

4.3.1 Compared with Self-adaptive Architecture for AUSs. PLASMA [73] is an architecture for a plan-based layered architecture for software model-driven adaptation. The architecture supports both reconfiguration and behavior adaptation. There is no goal management module in the system framework, and it is assumed that system goals are first addressed as a behavior problem and then an adequate reconfiguration is produced for the behavior strategy that is computed. The former is achieved through adaptation plans while the latter is achieved through application plans. The cascaded adaption plan generation process may result in sub-optimal goal achievement, which cannot guarantee critical ones like safety. In our architecture, `Safe Planner` is also allow for system reconfiguration (e.g., sensor reconfiguration) and behavior adaptation (motion planning) simultaneously. Additionally, the goal achievement is treated as a dynamic control problem to solve online, in which the critical safety requirements can be guaranteed.

MORPH [13] is an architecture for self-adaptive systems that involves runtime system reconfiguration and behavior adaptation. It allows for a coordinated yet transparent and independent adaptation of system configuration and behavior. In MORPH, precomputed behavior and reconfiguration strategies are selected to allow rapid adaptation. Although the behavior adaptation based on discrete event control theory can respond to environmental changes like obstacles, such a method may limit the capability and real-time responsiveness of AUSs, especially in a highly open and dynamic environment. To tackle this problem, we introduce the `Environment Monitor` and `Environment Situation Analyzer` for environment perception and prediction. Safe motion plans are automatically generated by `Safe Planner` to avoid potential intersections with obstacles, humans, or other systems dynamically and continuously.

4.3.2 Compared with Safe Architecture for AUSs. SAFECASS [45] is an architectural approach for developing surgical robot systems with safety in a reusable and structured manner. In SAFECASS, the system can proactively prevent errors by reacting to events that may initiate system state changes, while fault detection and diagnosis are realized based on a built-in filter for component-based robotic systems [44]. However, their work does not consider, as we do, how to ensure the safety of AUSs during their interaction with environmental entities (i.e., the feed-forward control loop).

SERA [28] is a software architecture to foster a fault-tolerant way of managing a team of robots through adaptation and coordination in a decentralized fashion. In SERA, fault-tolerance mechanisms are only designed for the fault that happens in the interaction between robots, hardware, or software. However, in a human-robot interaction scenario, there is a human-in-the-loop that can have potential safety threats system. In our architecture, we consider two types of human-robot interaction, i.e., humans as co-workers or the bystanders. In the case of humans as co-workers, AUS safe motions are decided either independently or cooperatively. In the other case, AUS is designed to observe and predict human motions and avoid collisions in advance.

In summary, existing safe architectures for the AUS mostly focus on specific robotic systems. Through SLR, we establish a taxonomy of safety concerns for the architectural design of the AUS, in the form of feature models which summarize system components, environment characteristics, internal/external safety threats, and countermeasures. Thus, our architecture is suitable for various types of AUSs like UAV, autonomous vehicle, UUV, etc., as well as the safety design of AUSs which is composed of multiple agents, such as a drone fleet.

5 CONCLUSIONS

AUSs are likely to flourish in the foreseeable future. However, they need to ensure the safety of themselves and the entities in the operating environment. Through a systematic review and investigation of the safety concerns for the AUS, we establish a taxonomy with six feature models of system components, environment characteristics, internal and external safety threats, and countermeasures. Based on this taxonomy, safety requirements are elicited, which drives the design of a safe architectural model for the AUS. It is our hope that this taxonomy for AUSs safety and reference architecture will provide the guidance for AUS safety design and motivate

the research community to develop new methods and techniques for AUS safety, making future AUSs safer in an autonomous way. In future research, we plan to apply the reference architecture to more practical scenarios to strengthen its applicability, such as autonomous vehicles. We will focus on applying robust and reliable AI-based safety measures in the `Safe Planner` of our architecture for more intelligent and self-adaptive AUSs.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61620106007 and 61751210. Zhi Jin is corresponding author.

REFERENCES

[1] Raja Ben Abdessalem, Annibale Panichella, Shiva Nejati, Lionel C Briand, and Thomas Stifter. 2018. Testing autonomous cars for feature interaction failures using many-objective search. In *Proc. 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 143–154.

[2] Flavia Sofia Acerbo, Herman Van der Auweraer, and Tong Duy Son. 2020. Safe and Computational Efficient Imitation Learning for Autonomous Vehicle Driving. In *Proc. American Control Conference (ACC)*. 647–652.

[3] Rachid Alami, Alin Albu-Schäffer, Antonio Bicchi, Rainer Bischoff, Raja Chatila, Alessandro De Luca, Agostino De Santis, Georges Giralt, Jérémie Guiochet, Gerd Hirzinger, et al. 2006. Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1–16.

[4] Daniel Althoff, Matthias Althoff, and Sebastian A. Scherer. 2015. Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3470–3477.

[5] Daniel Althoff, James J. Kuffner, Dirk Wollherr, and Martin Buss. 2012. Safety assessment of robot trajectories for navigation in uncertain and dynamic environments. In *Proc. IEEE International Conference on Robotics and automation (ICRA)*. 285–302.

[6] John D Andrews, J Poole, and Wen-Hua Chen. 2013. Fast mission reliability prediction for Unmanned Aerial Vehicles. *Reliab. Eng. Syst. Saf.* 120 (2013), 3–9.

[7] Civil Aviation Authority. 2010. CAP 722 Unmanned Aircraft System Operations in UK Airspace—Guidance. *Directorate of Airspace Policy* (2010).

[8] UK Civil Aviation Authority. 2012. Unmanned Aircraft System Operations in UK Airspace–Guidance. *Cival Aviation Publication* 722 (2012).

[9] Algirdas Avižienis, Jean-Claude Laprie, and Brian Randell. 2004. Dependability and its threats: a taxonomy. In *Building the Information Society*. 91–120.

[10] Osiris A Valdez Banda, Sirpa Kannos, Floris Goerlandt, Pieter HAJM van Gelder, Martin Bergström, and Pentti Kujala. 2019. A systemic hazard analysis and management process for the concept design phase of an autonomous vessel. *Reliab. Eng. Syst. Saf.* 191 (2019), 106584.

[11] Subho S Banerjee, Saurabh Jha, James Cyriac, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. 2018. Hands off the wheel in autonomous vehicles?: A systems perspective on over a million miles of field data. In *Proc. 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 586–597.

[12] Andrew J Barry, Anirudha Majumdar, and Russ Tedrake. 2012. Safety verification of reactive controllers for UAV flight in cluttered environments using barrier certificates. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. 484–490.

[13] Victor Braberman, Nicolas D'Ippolito, Jeff Kramer, Daniel Sykes, and Sebastian Uchitel. 2015. Morph: A reference architecture for configuration and behaviour self-adaptation. In *Proc. 1st International Workshop on Control Theory for Software Engineering (CTSE@SIGSOFT FSE)*. 9–16.

[14] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. 2007. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.* 80, 4 (2007), 571–583.

[15] Adam Bry and Nicholas Roy. 2011. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. 723–730.

[16] Kai-Yuan Cai, João W. Cangussu, Raymond A. DeCarlo, and Aditya P. Mathur. 2003. An Overview of Software Cybernetics. In *Proc. 11th International Workshop on Software Technology and Engineering Practice (STEP)*. 77–86.

[17] Krzysztof Czarnecki. 2018. On-road safety of automated driving system (ads)-Taxonomy and safety analysis methods.

[18] Jesimar da Silva Arantes, Márcio da Silva Arantes, Claudio Fabiano Motta Toledo, Onofre Trindade Júnior, and Brian C Williams. 2017. An embedded system architecture based on genetic algorithms for mission and safety planning with uav. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1049–1056.

[19] John Dahl, Gabriel Rodrigues de Campos, Claes Olsson, and Jonas Fredriksson. 2019. Collision Avoidance: A Literature Review on Threat-Assessment Techniques. *IEEE Trans. Intell. Veh.* 4, 1 (2019), 101–113.

[20] Ewen Denney, Ganesh Pai, and Ibrahim Habli. 2012. Perspectives on software safety case development for unmanned aircraft. In *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 1–8.

[21] Masao Dohi, Kazuya Okada, Ikuo Maeda, Shigetoshi Fujitani, and Toshihiro Fujita. 2018. Proposal of collaboration safety in a coexistence environment of human and robots. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. 1924–1930.

[22] Mahir Dursun and Ismet Cuhadar. 2018. Risk based multi criteria decision making for secure image transfer between unmanned air vehicle and ground control station. *Reliab. Eng. Syst. Saf.* 178 (2018), 31–39.

[23] Raj Gautam Dutta, Xiaolong Guo, Teng Zhang, Kevin Kwiat, Charles Kamhoua, Laurent Njilla, and Yier Jin. 2017. Estimation of safe sensor measurements of autonomous system under attack. In *Proc. 54th Annual Design Automation Conference (DAC)*. 46.

[24] John Edwards. 2019. Signal Processing Improves Autonomous Vehicle Navigation Accuracy: Guidance Innovations Promise Safer and More Reliable Autonomous Vehicle Operation. *IEEE Signal Process. Mag.* 36, 2 (2019), 15–18.

[25] Milan Erdelj, Enrico Natalizio, Kaushik R Chowdhury, and Ian F Akyildiz. 2017. Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Comput.* 16, 1 (2017), 24–32.

[26] Forbes. 2020. Tesla In Taiwan Crashes Directly Into Overturned Truck, Ignores Pedestrian, With Autopilot On. *https://www.forbes.com/sites/bradtempleton/2020/06/02/tesla-in-taiwan-crashes-directly-into-overturned-truck-ignores-pedestrian-with-autopilot-on/#33159ad758e5* (2020).

[27] Alessio Gambi, Tri Huynh, and Gordon Fraser. 2019. Generating effective test cases for self-driving cars from police reports. In *Pro. 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. 257–267.

[28] Sergio García, Claudio Menghi, Patrizio Pelliccione, Thorsten Berger, and Rebekka Wohlrab. 2018. An architecture for decentralized, collaborative, and autonomous robots. In *Proc. IEEE International Conference on Software Architecture (ICSA)*. 75–84.

[29] Michael Gerstmair, Alexander Melzer, Alexander Onic, and Mario Huemer. 2019. On the Safe Road Toward Autonomous Driving: Phase noise monitoring in radar sensors for functional safety compliance. *IEEE Signal Process. Mag.* 36, 5 (2019), 60–70.

[30] Julio Godoy, Ioannis Karamouzas, Stephen J Guy, and Maria L Gini. 2016. Moving in a Crowd: Safe and Efficient Navigation among Heterogeneous Agents.. In *Proc. 25th International Joint Conference on Artificial Intelligence (IJCAI)*. 294–300.

[31] Vinicius G Goecks, Gregory M Gremillion, Vernon J Lawhern, John Valasek, and Nicholas R Waytowich. 2019. Efficiently combining human demonstrations and interventions for safe training of autonomous systems in real time. In *Proc. 33rd AAAI Conference on Artificial Intelligence (AAAI)*. 2462–2470.

[32] Paula Gonçalves, José Sobral, and Luís Andrade Ferreira. 2017. Unmanned aerial vehicle safety assessment modelling through petri Nets. *Reliab. Eng. Syst. Saf.* 167 (2017), 383–393.

[33] Guardian. 2018. Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian. *https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe* (2018).

[34] Ángel Manuel Guerrero-Higueras, Noemí DeCastro-García, Francisco Javier Rodríguez-Lera, and Vicente Matellán. 2017. Empirical analysis of cyber-attacks to an indoor real time localization system for autonomous robots. *Comput. Secur.* 70 (2017), 422–435.

[35] Nikita Bhardwaj Haupt and Peter Liggesmeyer. 2019. A Runtime Safety Monitoring Approach for Adaptable Autonomous Systems. In *Proc. 38th International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*. 166–177.

[36] Daojing He, Yinrong Qiao, Sammy Chan, and Nadra Guizani. 2018. Flight Security and Safety of Drones in Airborne Fog Computing Systems. *IEEE Commun. Mag.* 56, 5 (2018), 66–71.

[37] Jeevith Hegde, Ingrid Bouwer Utne, Ingrid Schjølberg, and Brede Thorkildsen. 2018. A Bayesian approach to risk modeling of autonomous subsea intervention operations. *Reliab. Eng. Syst. Saf.* 175 (2018), 142–159.

[38] Nico Hochgeschwender. 2019. Adaptive Deployment of Safety Monitors for Autonomous Systems. In *Proc. 38th International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*. 346–357.

[39] Bin Hu and Peter Seiler. 2015. Pivotal decomposition for reliability analysis of fault tolerant control systems on unmanned aerial vehicles. *Reliab. Eng. Syst. Saf.* 140 (2015), 130–141.

[40] ISO10218. 2008. Robots for industrial environments—safety requirements. (2008).

[41] Michael Jackson. 2001. *Problem frames: analysing and structuring software development problems.* Addison-Wesley.

[42] Qazi Hamza Jan, Sascha Klein, and Karsten Berns. 2019. Safe and Efficient Navigation of an Autonomous Shuttle in a Pedestrian Zone. In *Proc. 28th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD).* 267–274.

[43] Zhi Jin. 2018. *Environment Modelling based Requirements Engineering for Software Intensive Systems.* Elsevier, Morgan Kaufmann Publisher.

[44] Min Yang Jung and Peter Kazanzides. 2012. Fault detection and diagnosis for component-based robotic systems. In *Proc. IEEE International Conference on Technologies for Practical Robot Applications (TePRA).* 1–6.

[45] Min Yang Jung and Peter Kazanzides. 2016. An architectural approach to safety of component-based robotic systems. In *Proc. IEEE International Conference on Robotics and Automation (ICRA).* 3360–3366.

[46] Min Yang Jung, Russell H Taylor, and Peter Kazanzides. 2014. Safety design view: a conceptual framework for systematic understanding of safety features of medical robot systems. In *Proc. IEEE international conference on Robotics and automation (ICRA).* 1883–1888.

[47] Staffs Keele et al. 2007. *Guidelines for performing systematic literature reviews in software engineering.* Technical Report. Technical report, Ver. 2.3 EBSE Technical Report. EBSE.

[48] Nancy Leveson. 2011. *Engineering a safer world: Systems thinking applied to safety.* MIT press.

[49] Nancy G Leveson. 1995. *Safeware. System Safety and Computers: A Guide to Preventing Accidents and Losses caused by Technology.* Addison-Wesley.

[50] Józef Lisowski. 2019. Sensitivity of Safe Trajectory in a Game Environment on Inaccuracy of Radar Data in Autonomous Navigation. *Sensors* 19, 8 (2019), 1816.

[51] Lantao Liu and Nathan Michael. 2016. An MDP-based approximation method for goal constrained multi-MAV planning under action uncertainty. In *Proc. IEEE International Conference on Robotics and Automation (ICRA).* 56–62.

[52] Lifeng Liu, Yingxuan Zhu, Tim Tingqiu Yuan, and Jian Li. 2020. Continuous Safe Learning Based on First Principles and Constraints for Autonomous Driving. In *Proc. 34th AAAI Conference on Artificial Intelligence (SafeAI@AAAI).* 170–177.

[53] Yixing Luo, Yijun Yu, Zhi Jin, and Haiyan Zhao. 2019. Environment-centric safety requirements for autonomous unmanned systems. In *Proc. 27th International Requirements Engineering Conference (RE).* 410–415.

[54] James T Luxhøj and Ahmet Öztekin. 2009. A regulatory-based approach to safety analysis of unmanned aircraft systems. In *Proc. 8th International Conference on Engineering Psychology and Cognitive Ergonomics (EPCE).* 564–573.

[55] Mathilde Machin, Jérémie Guiochet, Hélène Waeselynck, Jean-Paul Blanquart, Matthieu Roy, and Lola Masson. 2016. SMOF: A safety monitoring framework for autonomous systems. *IEEE Trans. Syst. Man Cybern. Syst.* 48, 5 (2016), 702–715.

[56] Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Vivian Weller. 2017. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI).* 4745–4753.

[57] Richard Melnyk, Daniel Schrage, Vitali Volovoi, and Hernando Jimenez. 2014. A third-party casualty risk model for unmanned aircraft system operations. *Reliab. Eng. Syst. Saf.* 124 (2014), 105–116.

[58] Daniel Meltz and Hugo Guterman. 2019. Functional Safety Verification for Autonomous UGV-Methodology Presentation and Implementation on a Full-scale system. *IEEE Trans. Intell. Veh.* 4, 3 (2019), 472–485.

[59] Yannik Nager, Andrea Censi, and Emilio Frazzoli. 2019. What lies in the shadows? Safe and computation-aware motion planning for autonomous vehicles using intent-aware dynamic shadow regions. In *Proc. International Conference on Robotics and Automation (ICRA).* 5800–5806.

[60] Samyeul Noh. 2018. Decision-Making Framework for Autonomous Driving at Road Intersections: Safeguarding Against Collision, Overly Conservative Behavior, and Violation Vehicles. *IEEE Trans. Ind. Electron.* 66, 4 (2018), 3275–3286.

[61] Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, Russ Tedrake, and John C Duchi. 2018. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *Proc. 31st Advances in Neural Information Processing Systems (NeurIPS).* 9827–9838.

[62] Ahmet Oztekin, Cynthia Flass, and Xiaogong Lee. 2012. Development of a framework to determine a mandatory safety baseline for unmanned aircraft systems. *J. Intell. Robotic Syst.* 65, 1-4 (2012), 3–26.

[63] The Washington Post. 2014. Fallen from the skies. *https://www.washingtonpost.com/wp-srv/special/national/drone-crashes/database/* (2014).

[64] Mazeiar Salehie and Ladan Tahvildari. 2009. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.* 4, 2 (2009), 14.

[65] Junaed Sattar and James J Little. 2014. Ensuring safety in human-robot dialog—A cost-directed approach. In *Proc. IEEE International Conference on Robotics and Automation (ICRA).* 6660–6666.

[66] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2017. On a formal model of safe and scalable self-driving cars. *CoRR* abs/1708.06374 (2017).

[67] Stephen L. Smith, Jana Tumova, Calin Belta, and Daniela Rus. 2010. Optimal path planning under temporal logic constraints. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 3288–3293.

[68] Ian Sommerville et al. 2011. *Software engineering.* Boston: Pearson,.

[69] Jonathan Sprauel, Andrey Kolobov, and Florent Teichteil-Königsbuch. 2014. Saturated path-constrained MDP: Planning under uncertainty and deterministic model-checking constraints. In *Proc. 28th AAAI Conference on Artificial Intelligence (AAAI).* 2367–2373.

[70] Roni Stern and Brendan Juba. 2017. Efficient, Safe, and Probably Approximately Complete Learning of Action Models. In *Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI).* 4405–4411.

[71] Selected Primary Studies. 2022. Selected primary studies for "A Taxonomy for Architecting Safe Autonomous Unmanned Systems". https://figshare.com/s/38ca3ec778c02928aec1

[72] Zachary N Sunberg, Mykel J Kochenderfer, and Marco Pavone. 2016. Optimized and trusted collision avoidance for unmanned aerial vehicles using approximate dynamic programming. In *Proc. IEEE International Conference on Robotics and Automation (ICRA).* 1455–1461.

[73] Hossein Tajalli, Joshua Garcia, George Edwards, and Nenad Medvidovic. 2010. PLASMA: a plan-based layered architecture for software model-driven adaptation. In *Proc. 25th IEEE/ACM International Conference on Automated Software Engineering (ASE).* 467–476.

[74] Christoph A Thieme and Ingrid B Utne. 2017. Safety performance monitoring of autonomous marine systems. *Reliab. Eng. Syst. Saf.* 159 (2017), 264–275.

[75] Cumhur Erkan Tuncali, James Kapinski, Hisahiro Ito, and Jyotirmoy V Deshmukh. 2018. Reasoning about safety of learning-enabled components in autonomous cyber-physical systems. In *Proc. 55th Annual Design Automation Conference (DAC).* 1–6.

[76] Federico Vicentini, Mehrnoosh Askarpour, Matteo G Rossi, and Dino Mandrioli. 2019. Safety assessment of collaborative robotics through automated formal verification. *IEEE Trans. Robotics* 36, 1 (2019), 42–61.

[77] Michael Vierhauser, Sean Bayley, Jane Wyngaard, Wandi Xiong, Jinghui Cheng, Joshua Huseman, Robyn R Lutz, and Jane Cleland-Huang. 2021. Interlocking Safety Cases for Unmanned Autonomous Systems in Shared Airspaces. *IEEE Trans. Software Eng.* 47, 5 (2021), 899–918.

[78] Inna Vistbakka, Amin Majd, and Elena Troubitsyna. 2018. Multi-layered Approach to Safe Navigation of Swarms of Drones. In *Proc. 37th International Conference on Computer Safety, Reliability, and Security (SAFECOMP).* 112–125.

[79] Akifumi Wachi, Yanan Sui, Yisong Yue, and Masahiro Ono. 2018. Safe exploration and optimization of constrained mdps using gaussian processes. In *Proc. 32nd AAAI Conference on Artificial Intelligence (AAAI).* 6548–6556.

[80] Yue Wang, Swarat Chaudhuri, and Lydia E. Kavraki. 2018. Bounded Policy Synthesis for POMDPs with Safe-Reachability Objectives. In *Proc. 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS).* 238–246.

[81] Yijing Wang, Zhengxuan Liu, Zhiqiang Zuo, Zheng Li, Li Wang, and Xiaoyuan Luo. 2019. Trajectory Planning and Safety Assessment of Autonomous Vehicles Based on Motion Prediction and Model Predictive Control. *IEEE Trans. Veh. Technol.* 68, 9 (2019), 8546–8556.

[82] Matt Webster, Michael Fisher, Neil Cameron, and Mike Jump. 2011. Formal methods for the certification of autonomous unmanned aircraft systems. In *Proc. 30th International Conference on Computer Safety, Reliability, and Security (SAFECOMP).* 228–242.

[83] Lora G Weiss. 2010. A network-based approach for assessing co-operating manned and unmanned systems (MUMS). In *Proc. 10th Performance Metrics for Intelligent Systems Workshop (PerMIS).* 222–226.

[84] Danny Weyns. 2019. Software engineering of self-adaptive systems. In *Handbook of Software Engineering.* 399–443.

[85] N. Yakymets, S. Dhouib, H. Jaber, and A. Lanusse. 2013. Model-driven safety assessment of robotic systems. In *Proc. IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS).* 1137–1142.

[86] Esen Yel, Tony X Lin, and Nicola Bezzo. 2018. Self-triggered Adaptive Planning and Scheduling of UAV Operations. In *Proc. IEEE International Conference on Robotics and Automation (ICRA).* 7518–7524.

[87] ZDNet. 2015. Google's autonomous car injuries: Blame the human. *https://www.zdnet.com/article/googles-autonomous-car-injuries-blame-the-human/* (2015).

[88] Man Zhang, Shaukat Ali, and Tao Yue. 2019. Uncertainty-wise test case generation and minimization for cyber-physical systems. *J. Syst. Softw.* 153 (2019), 1–21.

[89] Ruohan Zhang, Yue Yu, Mahmoud El Chamie, Behçet Açikmese, and Dana H Ballard. 2016. Decision-Making Policies for Heterogeneous Autonomous Multi-Agent Systems with Safety Constraints.. In *Proc. 25th International Joint Conference on Artificial Intelligence (IJCAI.* 546–553.

[90] Xunyu Zhong, Jun Tian, Huosheng Hu, and Xiafu Peng. 2020. Hybrid Path Planning Based on Safe A* Algorithm and Adaptive Window Approach for Mobile Robot in Large-Scale Dynamic Environment. *J. Intell. Robotic Syst.* (2020), 1–13.

[91] Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. 2009. Planning-based prediction for pedestrians. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 3931–3936.