The Safety of Autonomy: A Systematic Approach

John A. McDermid, **Radu Calinescu**, **Ibrahim Habli**, **Richard Hawkins**, **Yan Jia**, **John Molloy**, **Matt Osborne**, **Colin Paterson**, **Zoe Porter**, and **Philippa Ryan Conmy**, University of York

Autonomy does not subvert existing safety processes, but they must be supplemented with methods that address autonomy's challenges, especially where perception and decision-making tasks are implemented with machine learning. We present an approach to address the safety of autonomous systems, building on and complementing established safety engineering methods.

Traditionally, safety-related systems, like aircraft, cars, and factory robots, have operated under human control or supervision. With autonomous systems (ASs), the role of the human is lessened—perhaps just to the extent of initiating autonomous operation:

> An AS can operate independently of human control.

Digital Object Identifier 10.1109/MC.2023.3317329
Date of current version: 5 April 2024

ASs have existed for some time, for example, in rail, including the Docklands Light Railway in London. Although safety-critical, such systems operate within well-defined and, to an extent, controlled environments; for example, there are physical controls on human access to the tracks, and traffic movement is controlled through a signaling system. The introduction of such ASs has been successful, and they have good safety records.

By contrast, emerging ASs, such as autonomous vehicles (AVs) on the roads or collaborative robots (called *cobots*) in factories, operate in significantly more challenging

environments. We can refine the notion of autonomy by considering *context* and *purpose*:

> ASs operate within complex, dynamic, or partially observable contexts.
> ASs are intended to achieve objectives that can only be partially, or implicitly, defined.

Humans are generally good at dealing with open-domain and partially defined goals, such as driving from York to London, and will "fill in" the necessary details on the way. In developing an AS, decisions on such details are being *transferred* from human to machine[1]: computers are used to realize capabilities that humans have undertaken hitherto. Thus, from a safety perspective, the most fundamental challenge in assessing and assuring ASs is dealing with decision making by a machine that does not have the contextual and semantic "understanding" of the world that humans have (both taught and learned as we grow up).

A secondary, but very important, challenge is the use of machine learning (ML) for key functions of the AS, for example, to build situational awareness. With ML, a computer system is trained to carry out a task by exposing it to data that provide experience of undertaking that task. Thus, the objectives are not directly defined. The ML-based system can then perform the same task on data it has not seen before as it encounters new *contexts*, for example, a road segment for an AV or a patient for a robotic surgeon.

To summarize, the key safety challenges for ASs arise from their operation without human control in complex environments with partially or implicitly defined objectives, often using ML to realize critical system capabilities. However, ASs do not subvert established safety concepts and principles—for example, the need to search for and eliminate or mitigate single points of failure—but existing methods need to be complemented or revised to deal with these challenges. Thus, the aim of our article is to set out a systematic approach to dealing with the challenges of ASs that complements and supplements established safety engineering methods.

The article is organized as follows. The next section briefly considers related work. The sections "Conceptual Framework," "Safety of Autonomy in Complex Environments," and "Assurance of ML for ASs" describe the overall approach and assurance frameworks for ASs, and their ML elements, within a whole system context. The section "Discussion" outlines several uses of the approach, identifies broader issues that need to be addressed, and presents our conclusions.

RELATED WORK

Artificial intelligence (AI) safety is currently receiving a lot of attention, with a focus on nonphysical harms, such as bias in algorithmic decision making. Such issues are important; however, the safety of AS and ML, in the sense of physical harm, has received less attention, except by standards bodies. For example, Underwriters Laboratories published a standard for AS evaluation that is intended to be usable cross domain.[2] There are also domain-specific standards addressing particular problems, for example, the safety of the intended functionality (SOTIF)[3] in the automotive sector, which is concerned with showing that nominal behavior is safe and addresses some of the challenges of the partially or implicitly defined objectives identified previously. Generally, these standards are goal setting, although they do identify assessment "methods" in broad terms. There are other generic frameworks concerned with the safety of ASs, such as the Safety Critical Systems Club guidelines,[4] but these approaches generally fall short of objective criteria that could be used as a basis for compliance.

There is also a considerable body of work on the safety of specific aspects of ASs, such as reliable detection of pedestrians for AVs,[5] developing methods for hazard analysis on ML datasets,[6] and extending failure mode and effect analyses (FMEAs) to ML, including cybersecurity issues.[7] A few projects link traditional safety engineering and ML (see, for example, Jia et al.[8]), but there is generally a "gap" between the safety community and "nonsafety ML practice." Our article seeks to bridge this gap by setting out a systematic approach to bring the two together.

CONCEPTUAL FRAMEWORK

We represent an AS as operating cyclically using a sense–understand–decide–act (SUDA) model (this is conceptually similar to the approach proposed by Sifakis[9] and broadly equivalent to models like observe–orient–decide–act and monitor–analyze–plan–execute):

1. A system **senses** its environment to produce a representation of properties in the environment, which we will refer to as **sensed properties**. For example, a barometer can be used to measure atmospheric pressure.
2. A system **understands** its environment by analysis of the sensed properties to produce some more abstract representation of the environment, which we will refer to as **derived**

properties. For example, atmospheric pressure can be used to derive altitude. (The computation is nontrivial and typically involves compensation for meteorological factors, such as air temperature and areas of high and low pressure.)

The set of sensed and derived properties form a **world model**, which, in turn, gives the system **situational awareness**, that is, "knowledge" of its context. However, as noted previously, this may be complex, dynamic, and only partially observable.

3. A system then **decides** on a course of action to achieve its purpose.
4. Finally, a system **acts**, resulting in externally observable **behavior**, that is, a sequence of states and actions.

This behavior might include moving, switching on visible signals, such as navigation lights for a vessel, interaction with objects in the environment, such as picking soft fruit, or a drone broadcasting its position. These actions can change the state of the environment (context) as well as the AS itself, which is then sensed, leading to an evolution of behavior to continue to meet its purpose, such as picking all of the ripe raspberries.

This model of an AS provides the basis ("anchor") for our systematic approach to the safety and assurance of ASs, including their ML elements. Figure 1 shows that the sense–understand and decide–act elements of the AS can be implemented using conventional components, ML, or a combination thereof.

Two of these processes are concerned with developing safety cases (that is, structured arguments supported by evidence, intended to justify that a system is acceptably safe for a specific application in a specific operating environment)[10]:

- Safety of ASs in complex environments (SACE)[11]: concerned with the safety of the AS as a whole
- Assurance of ML component within an AS (AMLAS)[12]: concerned with the safety of the ML elements of an AS.

SACE also derives a safe operating concept (SOC), which is an abstract view of how the AS is intended to ensure safety. For example, for an AV this would include activities like specifying the SOTIF and requiring the AV to give way to emergency services. (However, recent events show that this does not always happen: https://www.nytimes.com/2023/08/18/technology/cruise-crash-driverless-car-san-francisco.html.) The SOC is defined for an operational domain model (ODM), which is the context in which the AS is developed to work safely (but note that the SOC must consider safe entry and exit from the ODM). (The term ODM is analogous to the term *operational design domain* used for AVs, but we chose a different term as the approach is intended to be more general and not restricted to just AVs.) The ODM helps to define the context for assuring the ML elements (using AMLAS) whether they are part of the "understand" or "decide" AS functionality.

Two other processes "bridge" between SACE and AMLAS:

- Safety assurance of understanding in ASs (SAUS)[13]: conducting a safety analysis of the sense and understand elements of the system, seeking to identify how errors could arise in the world model that could lead to hazardous decision making
- Safety assurance of decision making in ASs (SADA)[14]: conducting a safety analysis of the decision-making (and actuation) elements of the system, seeking to identify decisions that would be undesirable, even given a "perfect" world model.

Both of these processes help derive safety requirements for the system elements (SR_U and SR_D), given the AS-level safety requirements. (SR_U are safety requirements for understanding; SR_D are safety requirements for decision making, as illustrated in Figure 1.) For example, SAUS provides a modified form of hazard and operability study, which reflects the deviations from intent that can occur due to both sensing limitations, including environmental impairments, and the characteristics of the ML models used in the perception systems, which can lead to errors in the world model. SADA includes a decision safety analysis method that can be applied to the whole system (supporting SACE) and as a bridge between SACE and AMLAS.

In addition, SAUS and SADA provide specific verification and validation methods for the relevant system elements. Our focus here is on safety, but, for example, formal methods can be used in synthesizing adaptive controllers operating under uncertainty[15] as part of SADA.

The final process is the following:

- Social acceptability of ASs (SOCA)[16]: concerned with the overall acceptability of an AS in its social, not just physical or technical, context.

SOCA extends the ideas of safety cases to deal with ethics assurance. It brings

together the assurance case methodology with a set of ethical principles covering benefits, harms, human autonomy and fairness, or justice,[17] supplemented by transparency in response to the opacity of ML. Thus, SOCA provides a framework to reason about the broader ethical acceptability of an AS in its intended context. As shown in Figure 1, SOCA is interfacing to SACE; in this role it influences the definition of requirements for what is deemed acceptably safe. However, the intent is that SOCA will have a broader influence, for example, providing context for reasoning about bias in training data or identifying explainability requirements for ML-based tasks.

SAFETY OF AUTONOMY IN COMPLEX ENVIRONMENTS

The SACE assurance framework is illustrated in Figure 2.

SACE considers the AS holistically, starting with a concept definition and resulting in AS-specific arguments and evidence to complement "traditional" safety work, thus forming a complete system safety case. The intent is that the safety case will be developed to support a decision about whether to deploy the AS, which may involve regulatory scrutiny. The process consists of eight phases, with iteration over these phases as the design evolves, as emphasized by the "feedback and iterate" arrow. Not all of the phases explicitly use the term "assurance," but the overall intent is to provide assurance that the AS can operate safely, to achieve its *purpose*, in its

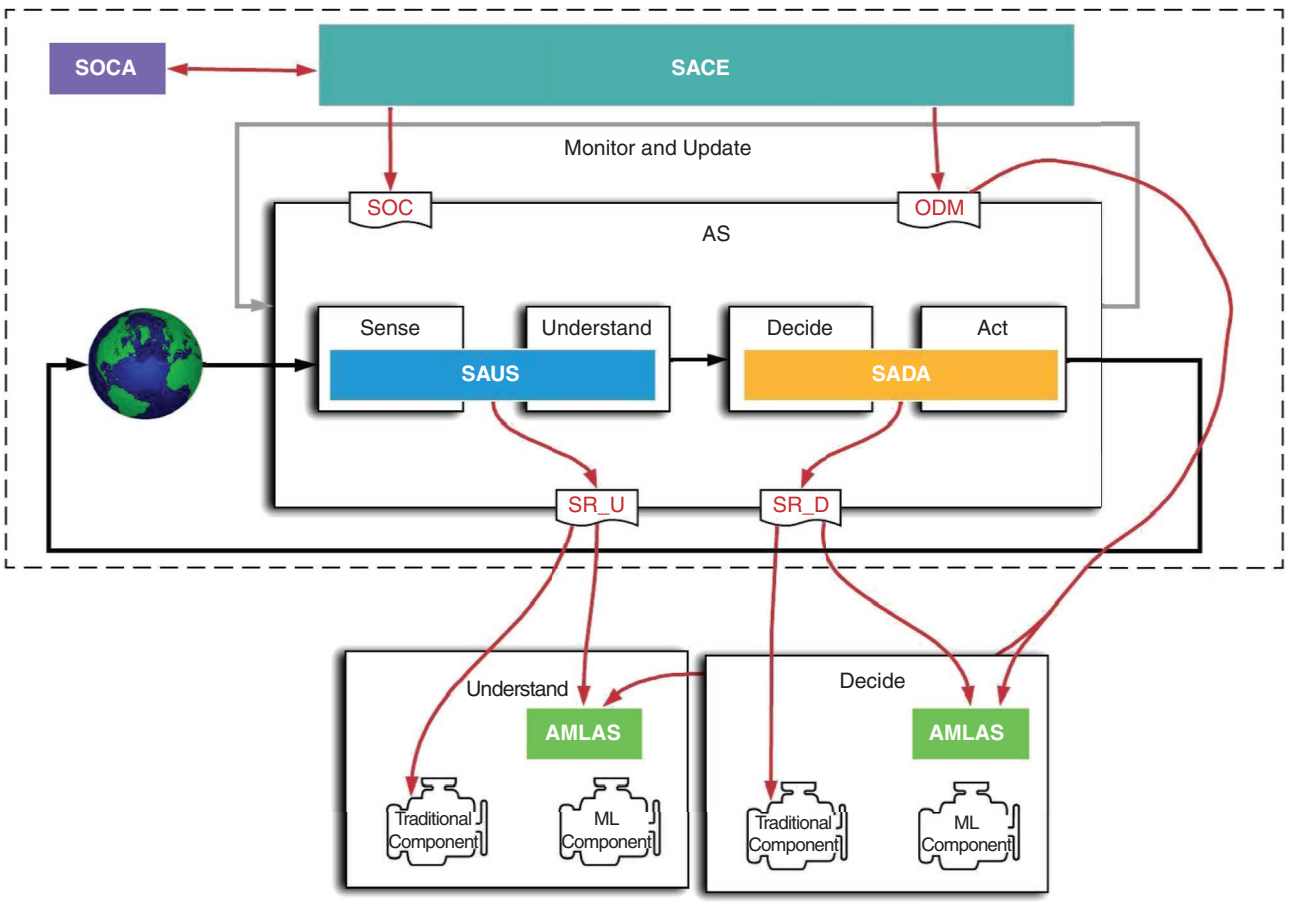

FIGURE 1. Safety and assurance processes for ASs. SOC: safe operating concept; SOCA: social acceptability of ASs; SACE: safety assurance of ASs in complex environments; ODM: operational domain model; SAUS: safety assurance of understanding in ASs; SADA: safety assurance of decision making in ASs; AMLAS: assurance of ML component within an AS.

context of use. Briefly, the concerns of each phase are as follows:

1. *Operating context assurance*: Understanding the context of operation for an AS, noting that safety is highly context dependent. For example, a humanoid robot used for assistive care in a retirement home may be physically the same as one used in a factory—but the hazards and mitigations are shaped by these different contexts. Hence, confidence is needed that the operating context has been defined appropriately.
2. *Identification of AS hazardous scenarios*: Because of the complexity of the operating context for many ASs, it is only realistic to identify hazards on a scenario-by-scenario basis; one of the concerns here is the sufficiency of such scenarios.
3. *SOC assurance*: The SOC must cover nominal behavior and off-nominal behavior, including challenging situations, such as sudden rainstorms for an AV or demanding sea states for an autonomous vessel. The SOC also needs to encompass safe responses to failures that may occur during the AS operation, such as loss of communications; see phase 6.
4. *AS safety requirements assurance*: Specifying and validating system-level safety requirements for the AS to realize the SOC (some of which will flow down to the ML elements, potentially refined by SAUS and SADA).
5. *AS design assurance*: Hierarchical and iterative decomposition, leading to the use of AMLAS for the ML elements but using conventional processes for the other AS elements.
6. *Management of hazardous failures*: Phases 1–5 are largely "top down," but this phase is "bottom up" and may include classical FMEAs while also having to deal with the effects of failures of ML elements, such as misclassifications or false negatives in object detection. This will influence the definition and assurance of the SOC; see phase 3.
7. *Out of context operation assurance*: It is unrealistic to assume that ASs always stay within the intended ODM, so this phase must address the AS, preserving safety outside the ODM where it is impracticable or unethical to hand control back to a human operator.
8. *AS verification and validation*: As for conventional systems, this includes scenario-based testing and use of simulation, but extended as necessary to deal with the characteristics of ML.

Each phase is broken down into a set of activities that take inputs, for example, the AS concept definition for phase 1, and produce a range of output artifacts, such as operating scenario definitions. This is illustrated in Figure 3 for

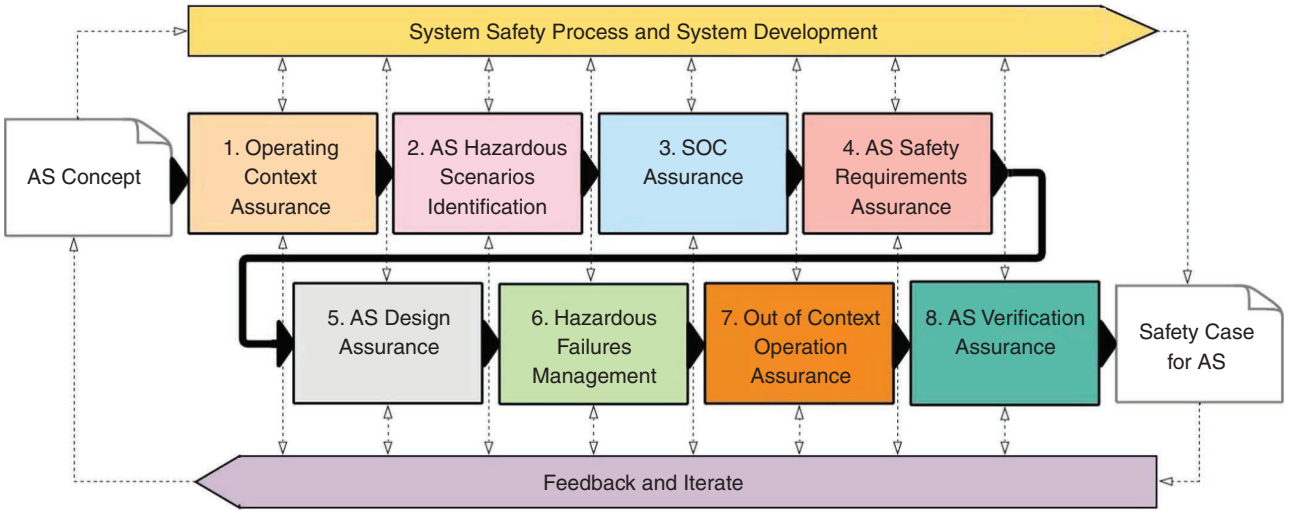

FIGURE 2. The SACE process.

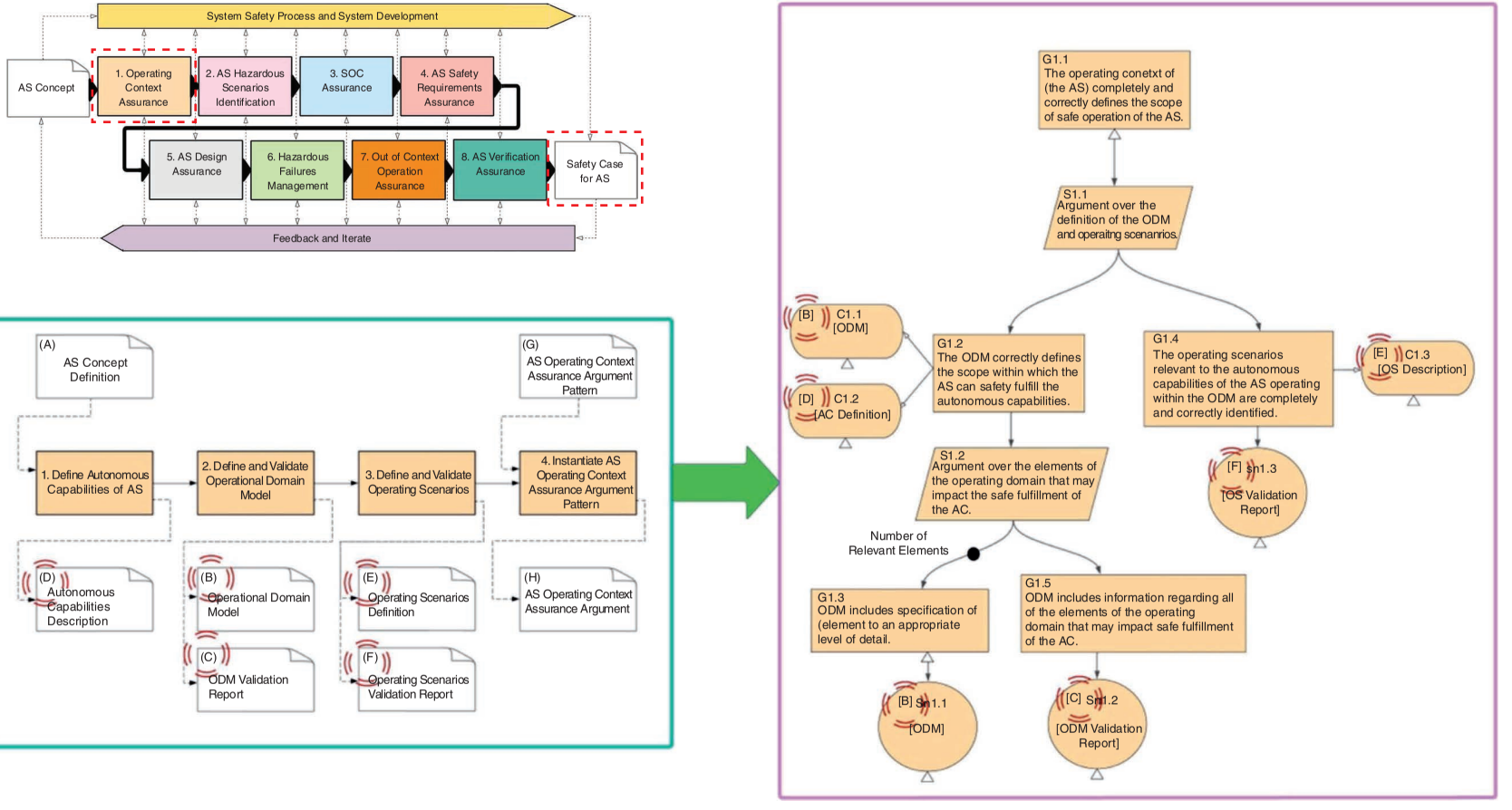

FIGURE 3. SACE operating concept assurance phase.

SACE phase 1. The last step for each phase takes a safety case argument pattern [defined using the goal structuring notation (GSN)[18]] and uses the generated artifacts to instantiate that pattern. The artifacts provide contextual elements (the lozenges) and solutions (the circles) in the instantiated safety argument. In Figure 3, artifact B (the ODM) is both contextual and a solution; artifact C (ODM validation report) is a solution only.

The whole of SACE is structured this way, with the argument pattern instantiations linking into a complete safety case. The SACE manual[11] gives guidance and examples on how to carry out the analysis and the scope of the output artifacts. SACE produces results to extend and complement traditional safety work and safety cases. For example, where autonomy and ML can contribute to existing hazards, the safety case will encompass definition of controls over the hazards/hazard causes (mainly phases 3–6 in SACE) and verification that these controls have been implemented (mainly phase 8). Further, autonomy may introduce new hazards, so the combined safety case would include the autonomy-specific hazards as well as more conventional ones. The out-of-context assurance is also likely to extend a conventional safety case—but link to it as, for example, loss of communication or propulsion for a mobile AS (conventional failure modes) may pose challenges when there are no humans present to manage the failure.

ASSURANCE OF ML FOR ASs

AMLAS[12] is structured similarly to SACE, and the safety case(s) produced using AMLAS can link into the AS-level safety case and thus into the wider system safety case. AMLAS is based on an extensive survey of the ML literature[19] and draws together safety processes and key ML concepts into a coherent framework. It is structured as follows, with phases 3–5 reflecting a typical ML development lifecycle:

1. *ML safety assurance scoping*: Defining context of use of an ML element of the AS; this is shaped by phase 1 of SACE but reflects the specific role of the ML element within the system, for example, object detection or path planning.
2. *ML requirements assurance*: Deriving specific performance and robustness safety requirements for the ML elements; the way this informs ML development (and assurance) will depend on the type of ML model used, for example, defining the proper reward function for reinforcement learning.[20]
3. *Data management assurance*: Concerned with the relevance, completeness, accuracy, and balance of the data, which depends on the *purpose* of the AS and the scope of the ML element; this is where concerns about bias would be addressed.
4. *Model learning assurance*: Concerned with showing that the learned model meets the safety requirements; this might, for example, include shaping the loss function[20] as well as explicit requirements-based testing.
5. *Model verification assurance*: This reflects the established ML development practice of "independent" testing of the ML model with a dataset that was not used in model development. Verification data particularly focus on realistic but challenging data that test the generalizability of the model within its context of use.
6. *Model deployment assurance*: This phase demonstrates that the properties demonstrated in the previous phases hold in the deployment context, that is, on the physical hardware, in the AS.

The AMLAS guidance[12] supports the systematic development of a safety case via instantiation of GSN patterns for the ML elements of the AS in a similar way to SACE.

DISCUSSION

As explained previously, the five elements of the framework build on established practices; for example, SACE reflects the systems and safety lifecycle models in Aerospace Recommended Practice 4754A.[21] Thus, the approach outlined complements traditional safety processes, especially those in regulatory contexts that employ safety cases to support deployment, as outlined when discussing SACE.

AMLAS and SACE are complementary, but there has been wider experience in using AMLAS so far. This experience spans multiple domains and is summarized next. (See https://www.york.ac.uk/assuring-autonomy/demonstrators/ for an illustration of the range of domains we have worked in. Some of this work predated AMLAS and SACE and helped to develop the concepts.) One example is the use of satellite-borne ML for wildfire detection, carried out in conjunction with Craft Prospect, a U.K.-based specialist satellite technology company. Here, AMLAS was used on the ML elements to demonstrate safety, which includes giving accurate predictions of the size and locations of wildfires, necessary for the safety of those fighting the fires.[22] An example of the independent use of AMLAS (involving none of the AMLAS

developers) was for the safety assurance of an emergency braking system for an AV, intended to protect pedestrians.[23] This can be seen as an example of assuring the SOTIF. These two examples are both embedded systems—the initial target of the framework—but AMLAS has also been successfully applied to decision support systems, for example, in healthcare.[24]

The work is also progressively influencing standards and regulations, for example, working with the British Standards Institution on standards in the aerospace, automotive, and healthcare sectors, and shaping health-care regulations.[25]

While there have been some initial successes in applying the approach, there remain limitations and issues of maturity. As with the standards, our approach is not precise about the level of evidence needed for the safety case—although both the definitions of SACE[11] and AMLAS[12] and the examples of the use of AMLAS[22,23,24] illustrate the approach and thus assist in interpreting and applying it. But this is work in progress, and there is more to be done.

Many emerging standards and guidelines for ASs introduce the notion of "levels" of autonomy, such as the Society of Automotive Engineers (SAE) levels for AVs[26] and the Grades of Automation in rail.[27] Our work does not yet address levels—and indeed it may not. Experience with AVs has cast some doubt on the utility of the concept, and we believe that—in alignment with safety engineering practice—it is better to focus on risk, rather than trying to simplify the approach using "labels." For example, systems of shared control (around SAE level 2+/3) may pose higher risks than level 4 or 5 because of the challenges regarding human supervision of autonomy and handover. We advocate an approach based on the analysis of function allocation between human and system[28] and see this as an enhancement of the SUDA system model. Other enhancements are needed; for example, it is also necessary for ASs to monitor their own health and to use such information to inform their decision making, such as changing their operational mode in the presence of sensor failures or impairment.

There are many other issues to be considered, including the wider concept of dependability, for example, sociotechnical[29] and availability concepts, not just safety. The long-established models of fault–error–failure[30] need reconsidering for AS and ML; there is a need to consider cybersecurity issues and the ability to maintain the safety of an AS through life. There are also opportunities to use ML in the support of safety engineering, for example, in automating the generation of FMEAs.[31]

Autonomy and the use of ML pose challenges for safety and assurance. What we have sought to do in this article is to illustrate how these challenges can be addressed by building on established safety engineering concepts but enhancing them with AS- and ML-specific approaches. There is much more to be done, but experience to date suggests that our systematic approach provides a basis on which we—and we hope the wider community—can build. However, we believe there is an urgent need for work to build consensus. In several domains, ASs are starting to become much more widely deployed (or there is pressure to deploy them), and localized regulations will lead to cost and complexity for developers and regulators alike and will not contribute to responsible innovation.

Finally, given the growing emphasis of "safe AI" on nonphysical harms arising from stand-alone AI systems, such as the mental health of users or discrimination against users, we believe that the combination of AMLAS[12] and our approach to ethics assurance arguments[16] might prove of wider value. C

ACKNOWLEDGMENT

We wish to acknowledge the support of the Lloyd's Register Foundation, the sponsors of the Assuring Autonomy International Programme since 2018.

REFERENCES

1. S. Burton, I. Habli, T. Lawton, J. McDermid, P. Morgan, and Z. Porter, "Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective," *Artif. Intell.*, vol. 279, Feb. 2020, Art. no. 103201, doi: 10.1016/j.artint.2019.103201.
2. *IEEE Standard for Evaluation of Autonomous Products*, UL Standard ANSI/UL 4600, 2022.
3. *Road Vehicles – Safety of the Intended Functionality*, ISO 21448, 2022.
4. "Safety assurance objectives for autonomous systems version 3," Safety of Autonomous Systems Working Group, Safety-Critical Systems Club, SCSC-153B, U.K., 2022. [Online]. Available: https://scsc.uk/scsc-153B
5. L. Gauerhof, R. Hawkins, C. Picardi, C. Paterson, Y. Hagiwara, and I. Habli, "Assuring the safety of machine learning for pedestrian detection at crossings," in *Proc. 39th Int. Conf. Comput. Saf., Rel., Secur. (SAFECOMP)*, Lisbon, Portugal: Springer-Verlag, Sep. 16–18, 2020, pp. 197–212.
6. H. Carter, A. Chan, C. Vinegar, and J. Rupert, "Proposing the use of hazard analysis for machine learning data sets," *J. Syst. Saf.*, vol. 58, no. 2,

ABOUT THE AUTHORS

JOHN A. McDERMID has been a professor of software engineering at the University of York, YO10 5GH York, U.K., since 1987 and twice served as head of the department. He is currently director of the Assuring Autonomy International Programme, focusing on the safety of robotics and autonomous systems, especially those employing machine learning. McDermid received a Ph.D. in computer science from the University of Birmingham. He is a Fellow of the British Computer Society, the Institute of Engineering and Technology, and the Royal Academy of Engineering, and is a chartered engineer. Contact him at john.mcdermid@york.ac.uk.

RADU CALINESCU is a professor of computer science at the University of York, YO10 5GH York, U.K., leads the Assuring Autonomy International Programme work on safety assurance of decision making in autonomous systems (ASs), and is the principal investigator on several projects on trustworthy ASs. His research focus is on the use of mathematical modeling and analysis to improve the safety and performance of self-adaptive systems and processes. Calinescu received a DPhil in computation from the University of Oxford. He is a member of the editorial board of Springer's *Computing* journal for the area of autonomic, adaptive, and dependable computing. Contact him at radu.calinescu@york.ac.uk.

IBRAHIM HABLI is a professor of safety-critical systems at the University of York, YO10 5GH York, U.K., the deputy head of the Department of Computer Science for research, and the director of research for the Assuring Autonomy International Programme. His research interests are in the design and assurance of safety-critical systems with a particular focus on intelligent systems and digital health. Habli received a Ph.D. in computer science from the University of York. Contact him at ibrahim.habli@york.ac.uk.

RICHARD HAWKINS is a senior research fellow in the Assuring Autonomy International Programme at the University of York, YO10 5GH York, U.K., where he is investigating the assurance and regulation of robotic and autonomous systems. Hawkins received a Ph.D. in computer science from the University of York. Contact him at richard.hawkins@york.ac.uk.

YAN JIA is a research fellow with the Assuring Autonomy International Programme at the University of York, YO10 5GH York, U.K. Her work focuses on safety management in health care, unifying and integrating work in traditional safety engineering and in machine learning. Jia received a Ph.D. in computer science from the University of York. Contact her at yan.jia@york.ac.uk.

JOHN MOLLOY is a research fellow with the Assuring Autonomy International Programme at the University of York, YO10 5GH York, U.K. His work includes the assurance of understanding and perception suites in autonomous systems (ASs) and the effects of environmental factors on sensor performance. Molloy received an EngD in applied photonics from Heriot-Watt University. Contact him at john.molloy@york.ac.uk.

MATT OSBORNE is a research fellow with the Assuring Autonomy International Programme (AAIP) at the University of York, YO10 5GH York, U.K., where he is currently researching software safety assurance in artificial intelligence and the safety of decision making in autonomy. Osborne received a PGCert in defense acquisition management from Cranfield University and is studying for a Ph.D. at the University of York. Contact him at matt.osborne@york.ac.uk.

COLIN PATERSON is a lecturer in computer science at the University of York, YO10 5GH York, U.K. His research focuses on how machine learning can be used safely in a range of contexts, including autonomous driving and the healthcare sector. He is particularly interested in uncertainty, the inadequacy of models, and their impact on decision making. Paterson received a Ph.D. in computer and control systems engineering from Coventry University and a Ph.D. in computer science from the University of York. Contact him at colin.paterson@york.ac.uk.

ZOE PORTER is a research fellow with the Assuring Autonomy International Programme (AAIP) at the University of York, YO10 5GH York, U.K., with a research focus on the ethics and governance of autonomous systems (ASs). Porter received her Ph.D. in philosophy from the University of York. Contact her at zoe.porter@york.ac.uk.

PHILIPPA RYAN CONMY is a research fellow with the Assuring Autonomy International Programme at the University of York, YO10 5GH York, U.K. She has extensive research experience in software assurance, certification, and safety cases. Conmy received a Ph.D. in computer science from the University of York. She is a Member of the British Computer Society and a chartered engineer. Contact her at philippa.ryan@york.ac.uk.

pp. 30–39, Jun. 2023, doi: 10.56094/jss.v58i2.253.
7. R. S. S. Kumar, D. O. Brien, K. Albert, S. Viljöen, and J. Snover, "Failure modes in machine learning systems," 2019, *arXiv:1911.11034*.
8. Y. Jia, T. Lawton, J. McDermid, E. Rojas, and I. Habli, "A framework for assurance of medication safety using machine learning," 2021, *arXiv:2101.05620*.
9. J. Sifakis, "Autonomous systems – An architectural characterization," in *Models, Languages, and Tools for Concurrent and Distributed Programming*, M. Boreale, F. Corradini, M. Loreti, and R. Pugliese, Eds. Cham, Switzerland: Springer-Verlag, 2019, pp. 388–410.
10. T. P. Kelly, "Arguing safety: A systematic approach to managing safety cases," Ph.D. dissertation, Univ. York, York, U.K., 1999.
11. "Guidance on the safety assurance of autonomous systems in complex environments (SACE)," Univ. York, York, U.K., 2022. Accessed: Aug. 22, 2023. [Online]. Available: https://www.york.ac.uk/assuring-autonomy/guidance/sace/
12. "Assurance of machine learning for use in autonomous systems (AMLAS)," Univ. York, York, U.K., 2021. Accessed: Aug. 22, 2023. [Online]. Available: https://www.york.ac.uk/assuring-autonomy/guidance/amlas/
13. J. Molloy and J. McDermid, "Safety assessment for autonomous system perception capabilities," 2022, *arXiv:2208.08237*.
14. M. Osborne, R. Hawkins, and J. McDermid, "Analysing the safety of decision-making in autonomous systems," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.*, Cham, Switzerland: Springer International Publishing, Jun. 2022, pp. 3–16.
15. R. Calinescu et al., "Discrete-event controller synthesis for autonomous systems with deep-learning perception components," 2022, *arXiv:2202.03360*.
16. Z. Porter, I. Habli, J. McDermid, and M. Kaas, "A principles-based ethics assurance argument pattern for AI and autonomous systems," *AI Ethics*, early access, Jun. 2023, doi: 10.1007/s43681-023-00297-2.
17. T. Beauchamp and J. Childress, *Principles of Biomedical Ethics*. New York, NY, USA: Oxford Univ. Press, 1979.
18. "Goal structuring notation community standard (Version 3)," Assurance Case Working Group, 2022. [Online]. Available: https://scsc.uk/r141C:1
19. R. Ashmore, R. Calinescu, and C. Paterson, "Assuring the machine learning lifecycle: Desiderata, methods, and challenges," *ACM Comput. Surv. (CSUR)*, vol. 54, no. 5, pp. 1–39, May 2021, doi: 10.1145/3453444.
20. Y. Jia, T. Lawton, J. Burden, J. McDermid, and I. Habli, "Safety-driven design of machine learning for sepsis treatment," *J. Biomed. Inform.*, vol. 117, May 2021, Art. no. 103762, doi: 10.1016/j.jbi.2021.103762.
21. *Guidelines for Development of Civil Aircraft and Systems*, SAE Int., Warrendale, PA, USA, SAE ARP4754A, 2010.
22. R. Hawkins, C. Picardi, L. Donnell, and M. Ireland, "Creating a safety assurance case for a machine learned satellite-based wildfire detection and alert system," *J. Intell. Robot. Syst.*, vol. 108, no. 3, Jul. 2023, Art. no. 47, doi: 10.1007/s10846-023-01905-3.
23. M. Borg et al., "Ergo, SMIRK is safe: A safety case for a machine learning component in a pedestrian automatic emergency brake system," *Softw. Qual. J.*, vol. 31, no. 2, pp. 1–69, Jun. 2023, doi: 10.1007/s11219-022-09613-1.
24. P. Festor, Y. Jia, A. C. Gordon, A. A. Faisal, I. Habli, and M. Komorowski, "Assuring the safety of AI-based clinical decision support systems: A case study of the AI Clinician for sepsis treatment," *BMJ Health Care Inform.*, vol. 29, no. 1, Jul. 2022, Art. no. e100549, doi: 10.1136/bmjhci-2022-100549.
25. S. Laher, C. Brackstone, S. Reis, A. Nguyen, S. White, and I. Habli, "Review of the AMLAS methodology for application in healthcare," 2022, *arXiv:2209.00421*.
26. "Taxonomy and definitions for terms related to driving automation systems for on- road motor vehicles," Society of Automotive Engineers (SAE), SAE-J3016, Sep. 2016. [Online]. Available: https://www.sae.org/
27. "Railway safety and interoperability in the EU," Publications Office of the European Union, European Union Agency for Railways, Luxembourg City, Luxembourg, 2022. [Online]. Available: https://www.era.europa.eu/content/railway-safety-and-interoperability-2022-report_en
28. H. E. Monkhouse, I. Habli, and J. A. McDermid, "An enhanced vehicle control model for assessing highly automated driving safety," *Rel. Eng. Syst. Saf.*, vol. 202, Oct. 2020, Art. no. 107061, doi: 10.1016/j.ress.2020.107061.
29. B. Townsend et al., "From pluralistic normative principles to autonomous-agent rules," *Minds Mach.*, vol. 32, no. 4, pp. 683–715, Dec. 2022, doi: 10.1007/s11023-022-09614-w.
30. A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan./Mar. 2004, doi: 10.1109/TDSC.2004.2.
31. D. Thomas, "Revolutionizing failure modes and effects analysis with ChatGPT: Unleashing the power of AI language models," *J. Failure Anal. Prevention*, vol. 23, pp. 911–913, May 2023, doi: 10.1007/s11668-023-01659-y.