Safety-Related Challenges for Autonomous Systems

The Safety of Autonomous Systems Working Group [SASWG]

January 2018

SCSC Publication Number: SCSC-143

Permanent URL: http://scsc.uk/scsc-143

The Safety Critical Systems Club (SCSC) is the professional network for sharing knowledge about safety-critical systems. It brings together: engineers and specialists from a range of disciplines working on safety-critical systems in a wide variety of industries; academics researching the arena of safety-critical systems; providers of the tools and services that are needed to develop the systems; and the regulators who oversee safety. Through publications, seminars, workshops, tutorials, a web site and, most importantly, at the annual Safety-critical Systems Symposium (SSS), it provides opportunities for these people to network and benefit from each other's experience in working hard at the accidents that don't happen. It focuses on current and emerging practices in safety engineering, software engineering, and product and process safety standards.

This document was written by the Safety of Autonomous Systems Working Group (SASWG), which is convened under the auspices of the SCSC. The goal of the SASWG is to produce clear guidance on how autonomous systems and autonomy technologies should be managed in a safety related context, throughout the lifecycle, in a way that is tightly focused on challenges unique to autonomy. The document was formally released at SSS'18, 6-8 February 2018.

Comments on this document are actively encouraged. These can be emailed to:

saswg-comments@scsc.uk

Contents

1 Introduction

This document aims to identify challenges associated with the demonstrably safe use of Autonomous Systems (AS), in order to scope the guidance provided by the Safety of Autonomous Systems Working Group (SASWG). The intent is to identify significant challenges, for example, those that: are applicable to a range of AS; are relevant to a range of different domains, or industrial sectors; are likely to require considerable effort to resolve.

1.1 Approach

Four complementary approaches have been used to elucidate relevant challenges:

- First, considering challenges from the viewpoint of different application domains. These considerations are based on the knowledge and experience of specialists working in each domain. They provide high-level themes and, as such, are well-aligned with the aims of this document. However, this perspective has the potential to miss items that would be uncovered by a more detailed, system-level consideration.
- Second, analysing a small number of fictitious, but plausible, example systems. This perspective provides a good description of specific issues, together with an appropriate supporting context. However, care needs to be taken to ensure that suitably generic observations are derived from these specific analyses.
- Third, recapping a previous publication that investigated the applicability of traditional software safety principles to the types of software likely to feature in AS. This perspective is valuable because software is almost always a critical component of an AS.
- Fourth, summarising a small number of accidents and incidents that have aspects relevant to AS. This perspective provides direct, real-world evidence of challenges. As with the earlier perspectives, care is needed when extracting generic challenges from specific issues. In addition, this perspective is inherently biased towards systems that have already been fielded and for which detailed accident (or incident) investigations have been undertaken.

No claim is made that any of these perspectives has been exhaustively examined. Likewise, no claim is made that the collection of perspectives is in any way complete. Despite these limitations, it is suggested that these perspectives are sufficient to support the further development of SASWG guidance. It is acknowledged that alternative approaches will be required to validate the guidance, once it has been developed.

This page is intentionally blank

2 Domain-Led Considerations

This section provides a brief discussion of key themes from the perspectives of particular domains; a brief summary of common themes is also provided. The following domains are considered:

- Air;
- Automotive;
- Defence;
- Maritime;
- Medical (Robot Surgery);
- Police and Criminal Justice;
- Space.

The intent is to use these discussions to help identify general challenges that are, in some way, widely applicable. As such, there is no particular need for any domain-specific section to be complete (in the sense that all possible challenges are identified).

Furthermore, many of the identified themes apply across several of the domains. For reasons of brevity, themes are only raised in one domain, even if they are applicable to several. This means that, whilst the combined contents of this section provide a useful perspective on potential AS-related challenges, it is inappropriate to draw conclusions from any single domain-specific section.

2.1 Air

Key themes related to the safety of autonomous systems in the air domain are discussed in the following paragraphs.

Focus of existing regulations There are detailed, well-understood and internationally-applicable regulations that cover the air domain; these are supported by detailed guidance material[1]. This body of information has been developed and used over a number of decades. As such, it is focussed on systems, and software, developed using conventional techniques. In particular, the regulations and guidance are not well-suited[2] to the type of software used in autonomous systems (eg, artificial intelligence, machine learning, etc). Developing mechanisms that can provide assurance of this software, along with convincing arguments as to the mechanisms' robustness, is a notable challenge.

Interface with Air Traffic Control Part of the air environment involves coordinating with Air Traffic Control (ATC) services. This currently involves verbal communication between humans although, in the longer-term, different ways of communicating may be implemented. However, it is likely that autonomous air systems will be required to engage with ATC services in the same way that a human pilot currently does. Even though the nature of these communications is significantly more structured than general conversation, being able to reliably use this mechanism could be challenging.

---

1 Examples include: Aerospace Recommended Practice 4754A, "Guidelines for Development of Civil Aircraft and Systems"; DO-178C, "Software Considerations in Airborne Systems and Equipment Certification".

2 Ashmore, R, Lennon, E (2017) Progress Towards the Assurance of Non-Traditional Software. In Developments in System Safety Engineering, ISBN 978-1540796288.

Third-party risks By its very nature, aviation involves flying over third parties. In many cases the third party may be unaware of the aviation activity; even if they are aware, it is generally not possible for them to insulate themselves from the activity and its potential consequences. This may mean that autonomous air vehicles may be subject to more intense regulatory scrutiny than vehicles in other domains. The nature of regulation applied to the air domain is also likely to result in detailed scrutiny. Providing assurance arguments that can withstand such scrutiny is a significant challenge.

Reliance on external systems Although it is not a strict necessity, autonomous air systems are likely to make significant use of Global Navigation Satellite System (GNSS) / Global Positioning System (GPS) services. However, in order to be demonstrably safe, they must be able to function safely in situations when GNSS/GPS in unavailable. For example, in such situations they may be required to be able to return to their originating location. Managing the transition from GNSS/GPS-present to GNSS/GPS-absent may be challenging. In addition, reliably navigating without GNSS/GPS and without an on-board human who is able to interpret visual cues could also be challenging.

Removal of human senses as health monitors Pilots tend to get acquainted with their aircraft. This often means they can detect subtle changes (eg, in engine tone, in control responsiveness) that may be precursors to hazardous situations. In this way the pilot's senses act as additional aircraft health monitors. However, the contribution they make to system safety is not explicitly acknowledged in safety arguments, nor is it quantified in safety engineering activities. Removal of humans from the cockpit, or at least specially-qualified and experienced humans, has the potential to lead to an unquantified risk. Demonstrating that this risk has been understood and mitigated is a considerable challenge.

2.2 Automotive

Key themes related to the safety of autonomous systems in the automotive domain are discussed in the following paragraphs.

Assuring driver readiness The SAE standard[3] defines a number of levels of autonomy for on-road vehicles. In some of these the human driver is required to provide a fall back option for conducting the dynamic driving task. For example, Level 3 autonomy, "conditional automation", involves the system monitoring the environment but requires the driver to be able to step in and take control. Providing means of ensuring the driver is able to safely achieve this is a notable challenge. Whilst this challenge is perhaps most severe for Level 3 autonomy, it also applies to cases where the autonomy only applies to certain driving modes (rather than covering all driving models).

There is a difference between planned handover (eg, as an AS approaches the end of a motorway) and unplanned, or emergency, handover. In that context, it is interesting to note that an ethics commission, appointed by the German Federal Government attempts to avoid this particular challenge by stating[4]: "*The software and technology in highly automated vehicles must be designed such that the need for an abrupt handover of control to the driver ('emergency') is virtually obviated.*"

Another way of avoiding the handover problem, and consequently the need to assure driver readiness, is to produce fully autonomous systems; this is the approach being adopted by Waymo[5]. This does, however, lead to a large number of engineering challenges, not least ensuring that the system can cope with *any*

[3] Surface Vehicle Recommended Practice, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, SAE J3016, September 2016.

[4] Federal Ministry of Transport and Digital Infrastructure, Ethics Commission: Automated and Connected Driving. Available from http://www.bmvi.de/SharedDocs/EN/publications/report-ethics-commission.pdf?__blob=publicationFile (retrieved 19 September 2017).

[5] https://waymo.com/safetyreport/.

situation it may find itself in.

Connectivity with other vehicles and the environment A variety of different levels of connectivity with other entities are possible. In the least connected case, each autonomous vehicle makes decisions as an isolated unit. Alternatively, a group of vehicles could share information, for example, to facilitate platooning on a motorway: in most cases such groups would be expected to form, reform and disperse organically. Additionally, a vehicle, or a group of vehicles, may obtain information from the infrastructure (eg, about road conditions, or the traffic situation). Being able to seamlessly switch between these different levels of connectivity presents a notable challenge.

Connectivity changing safety significance The consequences of an accident involving a single vehicle, whilst highly significant to those involved, are unlikely to be of societal significance[6]. Conversely, an accident involving a collection of connected vehicles could be highly significant; it could involve tens of deaths (or even more). This observation suggests that the AS could have different safety implications, depending on the prevailing situation. Managing this challenge is important.

Security informed safety Modern cars already contain a significant amount of computing resource; this will further increase with the introduction of more autonomy. Likewise, whilst there are currently many interfaces between in-vehicle computing and the outside world (eg, to support servicing, or to provide entertainment) additional levels of autonomy may lead to significantly more connections with the outside world. Each connection brings with it valuable system-level features, but it also has the potential to provide a new vector by which cyber-related effects (either malicious or inadvertent) may be introduced. Providing the requisite combination of security, safety and system functionality is a notable challenge[7].

Through-life behaviour monitoring All potential behaviours of an AS may not be known when it is introduced into service. One strategy for managing the associated residual uncertainty is to monitor performance in real time and analyse in a cloud. This may allow trends towards hazardous behaviour to be detected and preventative action to be undertaken. Such an approach will place integrity and assurance requirements on the collection, storage and processing of this data.

Behavioural updates A single autonomous car will experience many situations during a journey. Each situation has the potential to provide valuable data. Provided it was suitably assured, this data could be used to confirm that the current autonomous behaviour is acceptable; alternatively, or additionally, it could be used to refine the autonomous behaviour (either by learning new behaviours or tuning the parameters of existing ones). In the case when behaviour is refined, this learning could occur on-line (ie, within an individual vehicle) or data could be fed back to a cloud, for off-line learning with new software (and hence refined behaviours) being provided to the vehicle at a later date. Regardless of whether learning is on-line or off-line, there are several notable challenges associated providing behavioural updates, including:

- Ensuring that behaviours for recently experienced situations do not obscure, or remove, behaviours required for other situations. Essentially, this is about balancing long-term memory with short-term exploration.
- Achieving an appropriate balance between the pace of software updates and the assurance activities that are necessary to support an update. For example, on-line learning may require some form of on-line assurance process; alternatively, daily updates would be incompatible with an assurance process that took weeks to complete.

6 Note, however, that a single accident involving a new technology can, potentially, have societal significance. The incident relating to the use of Tesla's Autopilot (discussed in Section 5) is a possible example of this phenomenon.

7 The Department for Transport's "Principles of Cyber Security for Connected and Automated Vehicles" may be helpful in this regard: https://www.gov.uk/government/publications/principles-of-cyber- security-for-connected-and-automated-vehicles/the-key-principles-of-vehicle-cyber-security-for-connected-and-automated- vehicles (retrieved 12 September 2017).

- Managing the potential impact of apparently similar vehicles having different software versions and consequently exhibiting different behaviour. In this case there is a balance between combining learning from all vehicles and potentially introducing a common mode vulnerability.

Value of simulation Evidence from real-world driving will be an important part of the safety argument for any autonomous vehicle. However, it is infeasible to complete the number of miles of driving that would be required to demonstrate meaningful safety levels[8]. Consequently, evidence gained from simulation is very important. To be able to provide this evidence, the simulation needs to exhibit certain characteristics (eg, it needs to be suitably validated, it needs to be under configuration control, etc). In addition, consideration needs to be given to the cases (or scenarios) that will be investigated in the simulation. Some of these topics fall within the established discipline of "design and analysis of computer experiments", others are likely to be domain-specific. For example, there may be value in defining a minimum scenario list that should always be addressed; this could, perhaps, be viewed as being analogous to the standard crash tests to which all new vehicles are subjected. However, any such list would need to be defined and used with care in order to prevent systems being over-optimised against this list, rather than being designed against real-world situations.

Customisation and data-dependency It is possible that autonomous vehicle designers will choose to re-use large parts of a self-driving system across a variety of different types of vehicle. In addition, they may choose to enforce different behaviours (for example, travelling more slowly near a school bus) through the use of data parameters. Whilst they have advantages, both of these approaches potentially pose assurance-related challenges. The first approach may involve testing autonomous vehicle behaviour in a product line environment, which could be challenging due to the complex, integrated nature of the autonomous vehicle. The second approach may add another dimension to testing activities, which are already a large, multi-dimensional problem.

Risk acceptance, transfer and ethics Increasing levels of autonomy will result in changes to the level of risk experienced by vehicle users and others that share the same environment (eg, pedestrians). Although, overall, risk levels may be expected to decrease there is the potential for certain population classes to suffer increased risks; that is, there is a potential for some risk to be transferred. Whether this transfer (should one exist) is appropriate is an ethical question and one that the wider society needs to answer. Being able to provide information (eg, estimated accident rates) to support decisions of this nature is a notable challenge.

2.3 Defence

The defence domain may be viewed as a specialisation of the other domains: for example, it includes systems that operate in the air and maritime domains. Likewise, defence systems may be subject to civilian regulation, military regulation, or a combination of both. Despite these similarities, the defence domain also introduces some additional complexities, including: complex and contested environments; system elements that are deliberately designed to cause harm (eg, weapons); and the need to integrate autonomous systems across multiple domains (eg, air, land, maritime).

Key themes related to the safety of autonomous systems in the defence domain are discussed in the following paragraphs.

Mission In addition to safety aspects, completion of the allocated mission is important for military systems.

[8] Kalra, N, Paddock, S M (2016) Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? RAND Corporation. Available from https://www.rand.org/pubs/research_reports/RR1478.html. (retrieved 23 January 2018).

Non-completion of missions can have safety aspects in conflict situations. Military autonomous systems need to survive in hostile and complex environments with the ability to complete their tasks where possible. Mission issues can generally be treated in a similar way to safety.

Test, Evaluation, Verification and Validtion[9], and Trust Test, Evaluation, Verification and Validation (TEVV) is a critical element for building high assurance of autonomy. Autonomy of systems is increasing to provide fully functional, self-governing platform/entities that enhance the military operational capability, including teaming with human actors. This progression requires a significant increase in the trust of autonomous, self-governing systems. This issue with trust is not uncommon; industries that incorporate automation and autonomy with cyber-physical systems often struggle with acceptance of new technology, both by users of the technology and with respect to methods used to formally verify performance and safety.

2.4 Maritime

Key themes related to the safety of autonomous systems in the maritime domain are discussed in the following paragraphs.

Long communications paths Autonomous marine vessels could be operating at a significant distance from land. More generally, they could be operating at a significant distance from those responsible for their behaviour. These distances will lead to delayed and, possibly, intermittent communications. Making the system robust against these types of issue could be a challenge, although approaches currently used to communicate to vessels are likely to be of some assistance.

Limited monitoring infrastructure In some domains, autonomous systems and those who operate them may be able to exploit existing infrastructure. However, the nature of the maritime environment means that large swathes are without any formal monitoring infrastructure. This may mean that autonomous vessels need to exhibit a greater degree of self-reliance than is the case for other domains. Demonstrating that this reliance has been achieved, across a suitable range of operating conditions, is a notable challenge.

Weather In some cases it may be possible for an autonomous system to protect itself against adverse weather conditions: for example, aircraft may be able to fly around storms. Conversely, the combination of vehicle speed and size of weather pattern may mean that autonomous maritime vessels cannot rely on being able to avoid adverse weather conditions. Hence, the capability of these systems needs to be demonstrated in a wide range of weather conditions. This could be challenging.

Hostile adversaries Although the potential for hostile actions exists in many domains (eg, cyber attacks, car thefts, etc) marine piracy is a known, and potentially significant, risk. The way that an autonomous system responds to this risk could lead to a number of challenges. Generally speaking, if the autonomous system is able to take actions to frustrate potential attackers then there needs to be some way of distinguishing between attackers and, for example, authorised maintenance engineers or people in distress. Being able to reliably distinguish between these different categories is a significant challenge.

[9] US Department of Defense, Research & Engineering, Autonomy Community of Interest (COI) Test and Evaluation, Verification and Validation (TEVV) Working Group, Technology Investment Strategy 2015-2018, May 2015.

2.5 Medical (Robotic Surgery)

This section is constrained[10] to the robotic surgery part of the medical domain. It is heavily based on an e-print[11]. Key themes are discussed in the following paragraphs.

Positional uncertainties Involuntary movement of the patient can pose a risk during surgery. There are, essentially, two different ways this can be addressed. Firstly, the patient can be mechanically fixed in place, but this often requires invasive pins and frames, which demand additional pre-operation activities. Secondly, the autonomous system can adapt to patient movement, perhaps through the use of appropriately placed markers. Whilst both of these approaches are feasible from a general perspective, selecting and implementing the best approach for any given application is a considerable challenge.

Sensing fidelity An unexpected collision between a surgical tool and tissue has the potential to cause significant harm. Part of limiting this potential harm involves sensing the increased resistance when a tool stops moving through air and starts moving through tissue. Being able to achieve the required level of sensing fidelity is a significant challenge.

Unclear requirements An advantage of autonomous systems is that they do not need specific, highly-detailed requirements. Nevertheless, some form of requirement, or specification of desired behaviour, is needed. In the surgical domain, this is not always readily available. For example, there is no consensus on the acceptable range of knee-to-hip angles in knee replacement surgery. This makes it difficult to decide whether an autonomous system is behaving in an appropriate manner.

2.6 Police and Criminal Justice

Key themes related to the safety of autonomous systems in the police and criminal justice domain are discussed in the following paragraphs.

Use of advisory systems A key aspect of autonomous systems is machine learning software. This type of software can be used in advisory systems within the police and criminal justice sector, for example, to provide sentencing advice. Whilst they can be useful, advisory systems can also create uncertainty as to where responsibility actually lies. In most cases it is asserted that responsibility rests with the human being advised. However, in order for that assertion to be valid, the following need to hold:

- Firstly, the human needs to be able to come to a decision independently of the advisory system. In the case of sentencing, this can be based on experience of similar cases and, hence, is relatively straightforward. However, there may be other advisory systems where this is more challenging.
- Secondly, the human needs to be able to disagree with the advisory system without undue fear of negative consequences, should that disagreement lead to an undesirable outcome. Demonstrating that people have this freedom is likely to be very challenging.

Unitentional biases The behaviour exhibited by machine learning software will reflect the properties of the data with which it was trained. Indeed, this is a key attribute of this type of software. However, if there are unintentional biases in the training data, these will be carried through into the software's behaviour.

[10] This constraint is a reflection of the time available to, and expertise of, current SASWG members. It does not indicate that medical applications are of a lesser priority than other areas. The same caveat also applies for domains not included in the current edition of this document.

[11] Yip, Michael, and Nikhil Das. "Robot Autonomy for Surgery." arXiv preprint arXiv:1707.03080 (2017).

Detecting these biases is a challenging problem; correcting for them, should they be detected, is even more challenging[12].

2.7 Space

Key themes related to the safety of autonomous systems in the space domain are discussed in the following paragraphs.

Delayed, limited communication Communication times with space vehicles may be measured in minutes and bandwidths may be measured in kilobits per second. These factors mean that that exchanging information with autonomous space systems is a slow process. It also means there is no scope for a human to intervene in anything like real-time and hence they are unlikely to be able to act as some form of safety monitor. Finding ways of expanding system capabilities, whilst coping with communication limitations, is a notable challenge.

Environmental uncertainties There is often only limited, imperfect knowledge of the environment in which an autonomous space vehicle will be used. This can make it difficult to validate sensor readings. More generally, it can also make it difficult to understand the differences between the actual situation and the situation perceived by the autonomous system. Finding ways of managing this uncertainty may be a challenge.

2.8 Summary

The themes discussed in the separate domains above can be aggregated into the following four general challenges:

- ***Providing compelling evidence in the absence of suitable regulatory and guidance material.*** Relevant considerations include, but are not limited to: management of frequent software updates and / or on-line learning; specification of suitable requirements; use of data obtained from simulations; and the potential for risk transfer.
- ***Demonstrating interactions between autonomous systems and their operators are appropriate.*** Relevant considerations include, but are not limited to: managing the implications of removing human senses from on-board the platform; assuring the operator is ready to take over (if required); managing the implications of communication delays and limited bandwidth; and making appropriate use of advisory systems.
- ***Demonstrating interactions between autonomous systems and second, and third, parties are appropriate.*** Relevant considerations include, but are not limited to: engaging in verbal communication (eg, with ATC services); dynamically forming and reforming connections with other vehicles (including accounting for changes in safety significance); and protecting against hostile actors, both cyber and physical.
- ***Demonstrating autonomous systems have suitable behaviour in uncertain environments.*** Relevant considerations include, but are not limited to: the effects of GNSS/GPS unavailability; the impact of adverse weather conditions; difficulties in accurately sensing the environment; and uncertainty in characteristics of the actual environment.

[12] Albarghouthi, A., D'Antoni, L., Drews, S., and Nori, A. (2017). Quantifying Program Bias. arXiv preprint arXiv:1702.05437.

This page is intentionally blank

3 Fictitious Systems

This section contains relatively simple analyses of two fictitious example systems. Each analysis begins with an overview of the system; this is followed by assumptions made in the analysis; the main features relating to autonomous system safety are then discussed; and a closing summary is provided.

Note that the systems discussed in this section are not intended to withstand close technical scrutiny. They are only intended to provide a suitable context within which potential challenges can be highlighted.

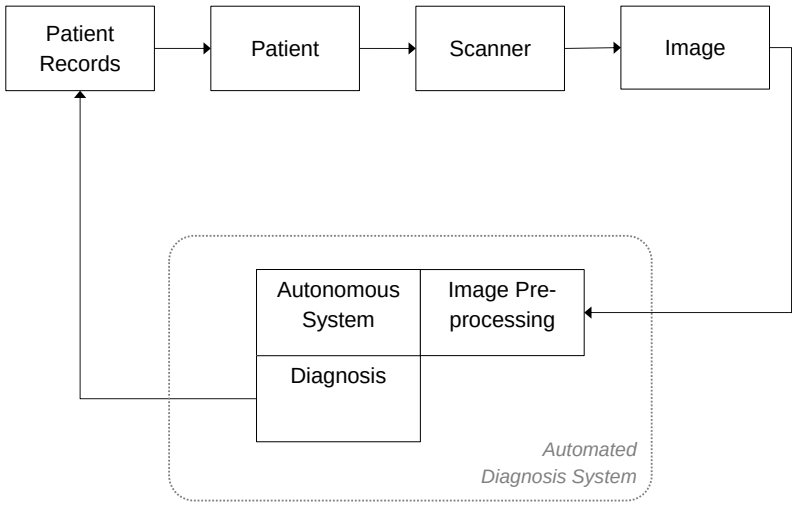
The following example systems are considered:

- A medical imaging system, which can make automated diagnoses;
- A self-parking car, which can park whilst unattended.

3.1 Medical Imaging System

3.1.1 Medical Imaging System - Overview

Below is a schematic of the process supported by the AS. In outline: a patient's records declare that an image-based diagnosis is required; the patient is scanned; the resulting image is fed to an automated diagnosis system, a component of which is the AS; the resulting diagnosis is recorded in the patient records.



For the purposes of this example, it is assumed that the diagnosis will be one of the following three[13] categories:

- Re-Image, which requests another image be captured (for example, because the supplied image is not of a high enough quality);
- Treatment, which initiates a course of treatment with unpleasant and potentially significant side effects;

[13] A potential fourth, non-diagnosis output, Barf, is described later.

- All-Clear, which suggests no further action need be taken.

3.1.2 Medical Imaging System - Assumptions

For the purposes of this analysis, the following assumptions are made:

- Issues associated with "patient records" and the "patient" (eg, ensuring the correct individual is scanned, ensuring the correct records are updated) are not related to the autonomous nature of the AS. Hence, whilst important, they are out of scope.
- Some pre-processing is likely to be required before the image is fed into the decision-making part of the AS. This pre-processing, which could be implemented using traditional software techniques, is not a focus for this analysis. It is, however, noted that this step offers a means by which the behaviour of the AS could be subverted[14].
- The task being performed by the AS is, in essence, image recognition. Hence, it is assumed that the AS is implemented using some form of Artificial Neural Network (ANN). That said, many of the assurance challenges would apply regardless of the implementation technology.
- It is assumed the ANN is developed via some form of training, test and validation activity, after which it is left fixed. In particular, when in operational use, providing the same image to the AS will always result in the same diagnosis[15]. (This assumption does not prevent observations from in-use behaviour being used to develop an improved AS.)

3.1.3 Medical Imaging System - Discussion

Consider, for example, a top-level claim that, "*The Autonomous System is suitably safe when used in the designed operating context*".

The key aspect[16] of "suitably safe" relates to whether the correct diagnosis has been provided. Note that there is no diagnosis that is always safe. For example, the AS diagnosing Re-Image rather than All-Clear wastes resources and causes unnecessary worry for the patient; if the image is an X-ray then this also leads to an increased risk due to exposure to repeated doses of radiation (similar considerations may also apply to other imaging techniques). Alternatively, diagnosing Re-Image rather than Treatment potentially delays treatment, which may lead to significant adverse effects.

It seems reasonable to rank the "badness" of at least some combinations of AS diagnosis and correct diagnosis. This is illustrated below, with larger values representing worse cases.

		Autonomous System Diagnosis		
		Re-Image	**Treatment**	**All-Clear**
Correct Diagnosis	**Re-Image**	0	(?)	(?)
	Treatment	2	0	4
	All-Clear	1	3	0

[14] Stevens, R., Suciu, O., Ruef, A., Hong, S., Hicks, M., and Dumitraş, T. (2017). Summoning Demons: The Pursuit of Exploitable Bugs in Machine Learning. arXiv preprint arXiv:1701.04739.

[15] Strictly speaking, it is the automated diagnosis system that produces the diagnosis. However, since the AS is the focus of this activity, within this document the term AS is used as a shorthand for referring to the wider diagnosis system.

[16] Other potential aspects include: always making a diagnosis; making a diagnosis within a specific time; etc.

This table indicates that, for example, the worst situation is where the AS returns an All-Clear, when the correct diagnosis would be Treatment. The cells marked "(?)" are cases where it is difficult to allocate a rank, principally because this depends on the correct diagnosis that would result from the Re-Image process.

Based on the approach illustrated above, the phrase "suitably safe" could be taken to mean: there is greater than an X% chance of a single use of the AS corresponding to a cell marked zero; and the cumulative chance of a single use corresponding either cell 3 or cell 4 is less than Y%. Obviously, suitable values for X and Y would need to be determined. If the intent is that the AS be at least as good as current practice then it seems reasonable to assume these values will be available.

Of course, this table relates to the performance of the system whilst in use. However, in use performance data is not available until sometime after the system has been fielded. This difficulty could be overcome by using the system on a trial basis in parallel with existing approaches, until sufficient data has been collected. Alternatively, the difficulty may be overcome by: firstly, assuring that the predicted performance satisfies the criteria in the preceding paragraph; and, secondly, providing sufficient confidence in the predictions.

Some aspects of predicting the performance of an ANN are well understood. Typically, these involve iterative splitting of a dataset into training and test data, with the process being supported by measures like precision and recall. However, some aspects of predicting performance warrant specific mention:

- Once the development team has produced a candidate ANN, this should be subject to independent validation. This involves a separate team analysing the performance of the ANN using data that is separate from that used by the development team for training and testing the ANN. Maintaining sufficient separation between the training and test data (used for development) and the validation data (used for independent validation) could be challenging. Furthermore, the validation data would be expected to contain cases specifically designed to fool the ANN[17], as well as cases that may result from failures in other parts of the process (eg, images corrupted either during capture by the scanner or in transmission to the ANN); determining the correct diagnoses for these cases could be challenging.
- In order to have confidence in the predicted performance of the ANN, confidence is needed in the Training, Test and Validation (TTV) data (noting that confidence in training and test data supports different types of argument than confidence in the validation data). For the example considered in this activity, this confidence could come from historical data that includes both the original image-based diagnosis and the final outcome; these could differ if, for example, treatment was subsequently required despite an initial All-Clear diagnosis.
- It is also important that the TTV data be suitably representative of the data that will be observed when the AS is in operational use. Formally speaking, the operational data should be statistically indistinguishable from the training and test data when viewed from the perspective of the input level neurons of the ANN. Note that the controlled nature of the process supported by the AS (eg, only images from medical scanners need analysing) means the operational environment of this AS is much simpler than that associated with, for example, an autonomous car[18].

In addition, to having confidence in predicted performance, there are other areas where confidence would be required. Examples include:

[17] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199.

[18] It may be possible to split the autonomous car's environment into a number of components each of which is no more complex than that of the AS considered here. An investigation into whether this environment splitting is feasible is outside the scope of the current activity.

- Confidence that typical errors associated with ANN development have been avoided. Some of these errors (eg, data leakage, overfitting) have been identified in previous work[19]. However, there does not appear to be an authoritative list of such errors. Furthermore, there do not appear to be standard ways of detecting and correcting them.
- Confidence that the software implementing the ANN has been developed using a suitable process. In theory, this should be straightforward; standard approaches for developing safety-critical software (for example, ones based on RTCA/DO-178C[20]) could be used. In practice, it could be difficult, for at least three reasons:
  - Firstly, the nature of ANN software may mean additional or alternative practices are required (eg, to account for the lack of explicit hierarchical decomposition of requirements);
  - Secondly, the organisation responsible for the ANN may not have a culture of developing "critical" software;
  - Thirdly, the ANN is likely to make use of externally-developed (and probably open source) libraries and frameworks, which are not under the control of the AS developer.

Finally, from the perspective of "suitably safe", there may be a need to implement some form of monitor on the output provided by the ANN. This could be achieved in at least two ways. For example:

- Requiring an appropriate fraction of images to be additionally analysed by a suitably qualified and experienced human. This could provide a system-level monitoring facility. Over the longer-term these results could be fed back into TTV data to produce an improved AS (should one be required).
- Implementing a collection of ANNs that exploit different underlying technologies (eg, different structures, different activation functions, etc). This is related to the concept of n-version programming for safety-critical software[21]. Similar to n-version programming, it could be difficult to quantify the value provided by multiple ANNs, especially if they are all based on the same collection of TTV data. Nevertheless, as a concept it offers the possibility of providing a per-image level of monitoring, and one that could take place in real time.

The preceding paragraphs have focussed on the "suitably safe" part of the suggested top-level claim that, "*The Autonomous System is suitably safe when used in the designed operating context*". The following paragraphs briefly consider the other part of the claim.

For the AS being considered in this activity a key part of the operating context is concerned with how the images seen in operational use relate to those in the TTV data. It has already been noted that, in order for there to be confidence in the ANN's predicted performance, the training and test data needs to be representative of the operational data. Although the "designed operating context" considers the same topic, it approaches it from a different perspective.

In particular, this context could restrict the type of scanner(s) that can be used with the AS. Likewise, it could also restrict the configuration settings of the scanner(s). It could even restrict their firmware versions. Ideally, this type of information (ie, scanner type, configuration settings, firmware version) would be supplied as metadata for the image. This would allow the AS to readily detect cases where it was

19 Hamner, B. (2014) Machine Learning Gremlins, in Strata Conference, O'Reilly, 11 Feb 2014.
20 RTCA (2011) Software Considerations in Airborne Systems and Equipment Certification, DO-178C, RTCA.
21 Knight, J. C., and Leveson, N. G. (1986). An experimental evaluation of the assumption of independence in multiversion programming. IEEE Transactions on software engineering, (1), 96-109.

being used in an inappropriate context[22]. In such cases it may be appropriate to produce a fourth output, colloquially referred to as Barf (ie, immediately exit with an appropriate message being displayed).

If suitable metadata cannot be made available, or if strict use of metadata is judged too restrictive, there may be a need for the ANN to determine whether it is being used in an inappropriate context. There does not appear to be a standard way of doing this. Potential options involve determining if an operational image is inside the convex hull of the TTV data and calculating the distance between the operational image and the closest element of TTV data, but neither of these is straightforward. Note, also, that there are inherent uncertainties involved in comparing a single image (ie, a single sample) with the collection of TTV data (ie, a statistical distribution).

3.1.4 Medical Imaging System - Summary

The relatively simple ANN-based AS considered in this activity raises several assurance challenges. Many of these relate to being able to confidently predict the performance of the AS, which also involves finding a way to express the desired performance of the AS. Other challenges relate to monitoring the AS whilst in use.

It is convenient to organise specific challenges against elements of the system life cycle, as indicated in the following table.

Life Cycle Element	AS-Related Challenge
Data	Demonstrating sufficient confidence that the TTV data is, firstly, correct and, secondly, suitably representative of the data that will be seen in operational use.
	Developing suitable validation data, which are sufficiently separate from the training / test data and which includes cases specifically designed to fool the ANN as well as cases that may result from failures in other parts of the wider process (eg, corrupted images).
Development	Demonstrating that typical errors associated with ANN development (eg, overfitting) have been avoided.
	Ensuring that suitable software development processes have been used, including for libraries or frameworks that the ANN is built upon.
Use	Confirming that an operationally-supplied image represents the AS being used in the designed operating context (eg, via metadata relating to the scanner and / or via a comparison between the operational image and the TTV data).
	Implementing an appropriate monitor (eg, a system-level monitor based on human observations, or a per-image level monitor based on multiple ANNs).

3.2 Self-Parking Car

3.2.1 Self-Parking Car - Overview

This AS represents a moderate extension on functionality that is available on a number of cars today. The system considered here is able to automatically park itself, with no need for a driver to be present in the vehicle. An illustrative scenario is outlined below:

[22] In addition to the factors already identified, the operating context could also conceivably constrain the training, qualifications and experience of the scanner operators.

- The driver navigates to a region where parking spaces may be found. These spaces may be of different types, for example: off-road car park; on-street parking; residential driveway; residential garage.
- The driver turns on the autonomous system, which searches for a suitable parking space whilst the driver navigates the immediate environment.
- If a space is found, the system presents this as an option to the driver. The driver performs their own check regarding the suitability of the space.
- If the space is deemed suitable, the driver stops the car, gets out and tasks the car to park itself.
- The car manoeuvres itself into the space and performs appropriate shutdown actions (eg, turning off the engine).

3.2.2 Self-Parking Car - Assumptions

For the purposes of this analysis, the following assumptions are made:

- When searching for a space, the system only considers physical accessibility; that is, whether the car can be safely manoeuvred into the space. It does not consider, for example, any restrictions that may apply to the space (eg, blue badge required, only available at certain times).
- If presented with sufficient information, the driver is able to determine whether a parking spot is appropriate. This includes, for example, confirming that there are no obstructions present. It also includes confirming that the vehicle will not cause a hazard if it cannot be extracted from the space (eg, as a result of a flat battery).
- The driver is able to confirm that the car is in a suitable state to be parked (eg, there are no passengers remaining in the vehicle).

3.2.3 Self-Parking Car - Discussion

As with the previous fictitious system, consider a top-level claim that, "*The Autonomous System is suitably safe when used in the designed operating context*".

Here, the term "suitably safe" includes preventing injuries to people and animals in the vicinity of the car. It also includes avoiding damage, either to the car itself, or other items in the environment (eg, other cars, road signs). Broadly speaking[23], being suitably safe can be summarised in two simple statements: firstly, do not hit anything; secondly, if something is hit, minimise the potential for injury or damage.

One important aspect of avoiding hitting anything involves the choice of parking space. As noted above, it is assumed that the driver is able to make a suitable determination if they are presented with suitable information. This places a requirement on:

- The way the information is gained. For example, if cameras are used then these need to be capable of providing sufficient resolution across different environmental (eg, lighting) conditions.

[23] There may also be a requirement to minimise disruption (eg, if parking on a busy road). For reasons of brevity, that is not considered in this example.

- The way information is presented. For example: if false colour overlays are used (eg, on non-optical sensors) then these need to be designed appropriately; if image processing techniques are used to identify (and subsequently highlight) items of special interest then the techniques need to be suitably assured.

Another aspect of avoiding hitting anything, the self-parking car needs to be able to detect[24] items in its immediate vicinity and, furthermore, it needs to be able to do this in a wide range of weather conditions and illumination levels. This is likely to demand a variety of different types of sensor; multiple instances of each sensor are also likely to be required (eg, to provide the necessary coverage around the vehicle and, possibly, to provide the required availability).

However, generally speaking, the self-parking car does not need to recognise, or classify, the objects it detects. It is sufficient to determine that there is something in the way[25]; there is no need to determine whether this is, for example, a child, a dog or a balloon. This significantly simplifies the way that the AS has to perceive the environment.

Another aspect of not hitting anything relates to the control algorithms used to manoeuvre the car. A large portion of these algorithms can be addressed using standard control theory. There are, however, some novel aspects that follow from the autonomous nature of the self-parking car. These are a consequence of removing the human, which has the effect of removing a safety monitor, from the vehicle.

Some safety monitoring aspects may be expected to be covered by the sensors used to detect objects. Other aspects could be covered by the monitoring items readily available on the car's internal network: for example, excessive speed, or excessive engine revolutions, could trip a safety monitor. Yet other aspects may require additional monitoring equipment, for example: audio sensors, to listen for unexpected sounds; accelerometers, to watch for unexpected movements of the steering wheel.

In terms of minimising the potential for injury or damage (should something be hit) a key consideration is the speed at which the vehicle manoeuvres; as discussed above, this should be relatively simple to implement. Another important consideration is the construction of the vehicle, for example, sharp edges ought to be avoided and materials that readily deform ought to be used; aspects like these are already considered in car design.

There are some special features of the "operating context" for this AS. For example, since the environment is one where cars are expected to park, it follows that it is one where low speed manoeuvring is possible, if not expected. As noted earlier, moving only at low speeds contributes to the second aspect of being suitably safe (ie, minimising the potential for injury or damage).

Another important feature of the operating context is that stationary cars should not pose a significant hazard. This means that a safe state for the self-parking car is simply to stop and turn on the hazard lights. Since it is expected to be moving slowly, it is readily apparent that the car should always be able to reach this safe state.

---

[24] Note that, due to the different ranges involved, the sensors used to perform these detections may be different to the sensors used to identify a potential parking space and present this as an option to the driver.

[25] For reasons of simplicity, this discussion does not address potential complexities like the self-parking car driving over a discarded crisp packet. It is, however, noted that an overly conservative approach, in which all objects were avoided, could produce a system that was safe, but unusable.

3.2.4 Self-Parking Car - Summary

Many aspects of safety for this autonomous system are already considered by current safety engineering processes; examples include: designing the car to reduce consequences should it hit a person; and implementing control algorithms to manoeuvre the car. Likewise, many aspects of manoeuvring the car into the parking space are well covered by existing control theory.

Although, initially, it may appear complex, the environment for this autonomous system is relatively simple. For example: there is a known safe state, which is readily reachable; the system only needs a basic representation of the environment (eg, the presence or absence of obstacles). These environmental aspects suggest it should be relatively easy to construct a safety argument.

4 Software Safety Principles

This section briefly summarises a paper[26] published at the 2017 SSS. The paper sought to apply the traditional "four plus one" software safety principles[27] to Machine Learning (ML) software; these types of software are expected to feature in a number of AS.

4.1 "Four Plus One" Principles

The following bullets provide a very brief overview of the "four plus one" software safety principles. A more detailed discussion of these principles is available in the original paper.

- **Principle One**

  Software safety requirements shall be defined to address the software contribution to system hazards.

  For the purposes of this principle, the software is treated as a black box: the principle is about ensuring there is a clear, documented understanding of what requirements the software needs to satisfy in order to provide a suitably safe system.

- **Principle Two**

  The intent of the software safety requirements shall be maintained throughout requirements decomposition.

  This principle acknowledges that there needs to be a process by which the system-level requirements (established in the previous principle) are developed into something that software can be implemented and tested against. The principle's text makes it explicit that this requirements development process involves decomposition: often, several stages are used (eg, moving through high-level software requirements to low-level software requirements), with traceability being maintained at each stage.

- **Principle Three**

  Software safety requirements shall be satisfied.

  This principle is about showing that the implemented software satisfies its requirements. (The previous two principles should ensure that these requirements are what is needed to provide a suitably safe system.)

- **Principle Four**

  Hazardous behaviour of the software shall be identified and mitigated.

  The first three principles provide assurance that the software meets the system-level requirements; the fourth principle is about showing that the software has not introduced any new hazards. This generally involves the application of a systematic process, for example, a Hazard and Operability Study (HAZOP).

- **Principle Four Plus One**

  The confidence established in addressing the software safety principles shall be commensurate to the contribution of the software to system risk.

  This principle acknowledges that, firstly, dissimilar pieces of software can pose different levels of system risk and, secondly, the available resource will inevitably be constrained (at least to some

[26] Ashmore, R, Lennon, E (2017) Progress Towards the Assurance of Non-Traditional Software. In Developments in System Safety Engineering, ISBN 978-1540796288.

[27] Hawkins, R, Habli, I, Kelly, T (2013) The principles of software safety assurance. 31st International System Safety Conference, Boston, Massachusetts USA.

degree). Hence, this principle helps developers distribute their resources across different parts of the system in an appropriate manner.

4.2 Application to ML Software

The main conclusions from the paper published at the 2017 SSS are:

- The "four plus one" software safety principles provide a good structure against which evidence for the use of ML software in safety-related systems may be judged. There are, however, two places where the structure could be enhanced:
  - Principle 2, "*the intent of the software safety requirements shall be maintained throughout requirements decomposition*", could be altered to reflect the lack of hierarchical decomposition in the production of ML software. For example, Principle 2' (two-primed) could read, "*the software detailed design shall embody the intent of the software safety requirements*".
  - To account for the potential impact of system adaptations, a new overarching principle may be appropriate: Principle 4+2, "*software required to produce behaviour not predictable at design time should consider the consequence of behavioural adaptations on its environment*".
- In order to apply the enhanced principles, further work is needed in at least two areas:
  - Demonstrating, without relying on a hierarchical structure, that intent has been maintained when moving from system-level software requirements to planned software implementation, as embodied in the detailed design (Principle 2 / 2').
  - New interpretations of HAZOP guide words tailored for ML software (Principle 4, "*Hazardous behaviour of the software shall be identified and mitigated*").
- The software verification philosophy embodied in DO-178C[28] is a helpful, if incomplete, framework for considering the verification of ML software. Additional work is required in a number or areas, including:
  - Understanding typical errors and how to avoid them, so as to inform robustness testing.
  - Ways of demonstrating that testing activities have been sufficiently complete.

4.3 Summary

Overall, the "four plus one" software safety principles provide a good, but incomplete, structure against which evidence required to support the operational use of ML software can be organised and discussed.

[28] RTCA (2011) Software Considerations in Airborne Systems and Equipment Certification, DO-178C, RTCA.

5 Accidents and Incidents

This section provides a brief summary of accidents and incidents that have aspects relevant to AS. To be clear, not all of the systems involved in these accidents and incidents would be considered to be AS. Nevertheless, it is judged that there is something to learn from the cases considered here that is relevant to AS safety.

Note: The analysis presented here has no legal standing whatsoever. The purpose of this section is not to discredit, contradict or undermine any existing accident analysis; the aim is simply to view historical occurrences from an AS perspective. In each case, a reference is provided to supporting material. All references have been taken at face value and not independently verified.

5.1 Loss of Hermes 450 Unmanned Air System (UAS), October 2011

5.1.1 Sources

- Service Inquiry investigating the accident involving Unmanned Air System (UAS) Hermes 450, ZK515 on 2 October 2011. https://www.gov.uk/government/publications/service-inquiry-investigating-the-accident-involving-unmanned-air-system-uas-hermes-450-zk515-on-02-oct-11 (retrieved on 13 June 2017).

5.1.2 Summary

On 2 October 2011 a Hermes 450 mission was terminated early due to increasing engine temperature. Transit back to the landing area was accomplished without incident. The Unmanned Air System (UAS) crew elected for an automatic landing using the GPS Take Off and Landing System (GTOLS).

The UAS self-aborted on approach due to an incorrect parameter in the GTOLS set-up that was loaded by the crew; this incorrect parameter was assessed as a contributing factor to the accident.

The crew elected to intervene rather than let the UAS self-recover from the abort. This intervention was against the written procedures and was assessed as an aggravating factor. Ultimately the aircraft hit a new, unoccupied United States Marine Corps hangar; the aircraft was subsequently assessed as having suffered Category 5 damage; ie, it was “non-repairable”.

5.1.3 Aspects Relevant to AS

The preceding summary raises two relevant aspects. Firstly, the importance of an incorrect parameter: in addition to hardware and software, AS are heavily dependent on data; equivalently, data safety is an important component of the safety of autonomous systems[29]. Secondly, the crew’s intervention as the aircraft self-recovered: this illustrates challenges in understanding (and trusting) AS behaviour in off-nominal situations, as well as the potential dangers of direct human intervention in AS behaviour (especially in off-nominal situations).

[29] The work of the SCSC’s Data Safety Initiative Working Group (DSIWG) is highly relevant to this concern.

5.2 Tesla Autopilot, May 2016

5.2.1 Sources

- The Automatic Emergency Braking (AEB) or Autopilot systems may not function as designed, increasing the risk of a crash, PE 16-007, 19 January 2017. https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.PDF (retrieved on 13 June 2017).
- National Transportation Safety Board, Public Meeting of September 12, 2017, Collision between a Car Operating with Automated Vehicle Control Systems and a Tractor-Semitrailer Truck, Williston, FL, May 7, 2016, NTSB/HAR-17-XX. https://www.ntsb.gov/news/events/Documents/2017-HWY16FH018-BMG-abstract.pdf (retrieved on 14 September 2017).

5.2.2 Summary

On 7 May 2016 a Tesla Model S collided with a tractor trailer crossing an uncontrolled intersection on a highway. The Tesla driver suffered fatal injuries. Data obtained from the Model S indicated that: the Tesla was being operated in Autopilot mode at the time of the collision; the Automatic Emergency Braking (AEB) system did not provide any warning or automated braking for the collision event; and the driver took no braking, steering or other actions to avoid the collision.

The examination by the US National Highway Traffic Safety Administration (NHTSA) examination did not identify any defects in the design or performance of the AEB or Autopilot systems of the subject vehicles. AEB systems are rear-end collision avoidance technologies that are not designed to reliably perform in all crash modes, including crossing path collisions. The Autopilot system is an Advanced Driver Assistance System (ADAS) that requires the continual and full attention of the driver to monitor the traffic environment and be prepared to take action to avoid crashes.

5.2.3 Aspects Relevant to AS

This accident raises several relevant aspects, including:

- The considerable amount of data that was available to support this investigation: this has the potential to significantly improve the "learning from incidents" part of safety culture (especially in non-aviation domains). A related recommendation from the US National Transport Safety Board (NTSB) is that the US Department of Transportation "*define the data parameters needed to understand the automated vehicle control systems involved in a crash*". The NTSB also recommend that the NHTSA, "*define a standard format for reporting automated vehicle control systems data, and require manufacturers of vehicles equipped with automated vehicle control systems to report incidents, crashes, and vehicle miles operated with such systems enabled*".
- The inadvertent, or deliberate, misuse of a driver aid (or, more generally, a supportive AS): this is partly about ensuring that people interacting with the AS are aware of its limits; it is also partly about taking positive action to ensure those limits are respected. A related recommendation from the NTSB to the NHTSA is that they develop a method to verify that "*vehicle automation systems incorporate system safeguards that limit the use of automated vehicle control systems to those conditions for which they were designed*". The NTSB also recommend that vehicle manufacturers "*incorporate system safeguards that limit the use of automated vehicle control systems to those conditions for which they were designed*".

5.3 Operator's Choice Overridden by Software, pre-1999

5.3.1 Sources

- Section F.6 of Software System Safety Handbook, Joint Software System Safety Committee, December 1999. http://www.system-safety.org/Documents/Software_System_Safety_Handbook.pdf (retrieved 13 June 2017).

5.3.2 Summary

During field practice exercises, a missile weapon system was carrying both practice and live missiles. Transit time was being used for slewing practice. Practice and live missiles were located on opposite sides of the vehicle.

The operator acquired the willing target, tracked it through various manoeuvers, and pressed the weapons release button to simulate firing the practice missile. Without the knowledge of the operator, the software was programmed to override his missile selection in order to present the best target to the best weapon.

The software optimized the problem, de-selected the practice missile and selected the live missile. When the release command was sent, it went to the live missile. The "friendly" target had been observing the manoeuvers of the incident vehicle and noted the unexpected live launch. Fortunately, the target pilot was experienced and began evasive manoeuvers, but the missile tracked and still detonated in close proximity.

5.3.3 Aspects Relevant to AS

It is perhaps surprising to consider an incident reported in a pre-2000 publication as being relevant to AS, especially as the system in question would probably be viewed as automatic, rather than autonomous. However, the SASWG has deliberately chosen not to attempt to precisely define the boundary between these two system classes.

In addition, this incident has characteristics that are interesting from the perspective of AS: firstly, the operator did not fully understand the behaviour of the software system; secondly the system developers (and testers) apparently failed to consider all potential scenarios (ie, they did not foresee any situation in which the "best" weapon should not be launched).

5.4 The Sterling "Flash Event", 2016

5.4.1 Sources

- Markets Committee. (2017). The sterling "flash event" of 7 October 2016 http://www.bis.org/publ/mktc09.pdf (retrieved 12 September 2017).

5.4.2 Summary

On 7 October 2016, in early Asian trading sterling depreciated by about 9% against the dollar, before quickly regaining much of the loss. Three phases can be identified within this event: rapid depreciation, in an orderly manner, due to significant selling; a number of minutes of extreme dysfunction, where sterling

traded over a large range; a gradual recovery in market liquidity and return to normality.

This event was the result of a combination of factors, including the behaviour of automatic trading algorithms as the currency traded through key levels and the fact that trading was being conducted outside of sterling's core time zone (so there was less liquidity and traders were less experienced in the market).

An investigation into the event concluded that this flash event was not a new phenomenon; instead it was another example of an event that has been observed across a broad range of fast, electronic markets. The investigation also notes, "*there is still a relatively limited understanding of the implications of widespread automated trading*".

5.4.3 Aspects Relevant to AS

Even though it relates to the financial sector, and includes relatively simple trading algorithms (rather than complex autonomous systems), this incident still illustrates an aspect relevant to the safety of autonomous system. In particular, this incident shows how the unplanned combination of algorithms can lead to undesirable behaviour.

Continuing that theme, it is conceivable that the combination of self-driving cars from different manufacturers could have negative system-level consequences (eg, traffic jams, increased accident rate) even though each car, when considered in isolation, was considered suitable for use.

5.5 Summary

The incidents outlined above highlight the following aspects of AS safety:

- **The importance of data** AS are likely to consume large amounts of data (eg, to support their training). Finding ways of confirming that all of this data is suitable for use may be difficult.
- **The importance of human-AS teaming** It is important that human operators understand the limitations and behaviour of an AS; this is especially important in off-nominal (or emergency) situations, which may exhibit limitations and behaviours that are not typically observed.
- **The importance of predicting all relevant situations** One advantage of AS is their ability to operate in situations that have not been explicitly considered in their design. However, there needs to be confidence that all relevant scenario variations have been considered; equivalently, assumptions used to constrain potential situations (eg, the same missiles will be fitted on each wing) need to be documented, agreed and confirmed.
- **The importance of considering AS-AS interactions** From one perspective, AS-AS interactions may be considered a special case of an individual AS interacting with its environment. However, the speed at which they can occur, and the potential for AS to behave differently from similar manned systems, means AS-AS interactions may need special consideration. This could be particularly challenging if the different AS are developed by competing organisations.

6 Generic Challenges

The preceding sections have used different perspectives to elicit challenges related to the demonstrably safe use of AS. It is again emphasised that none of these perspectives were intended to be complete (ie, cover every possible eventuality). Consequently, the summary of generic challenges presented in this section is also not intended to be complete.

At a top level, four main challenges can be identified. Any safety case for an AS will need argue convincingly that these challenges have been met.

6.1 It is Much Harder to Achieve Safety Of The Intended Function

In AS development, many of the functions that need to be specified (and then built, ensured and assured) are much more complicated than those used in current safety-critical systems. They will operate an environment of very high irreducible complexity, which leads to very high uncertainty about what behaviour would indeed be safe in context. They thus present an extremely difficult validation problem.

In particular, the gestalt whole-system behaviours (eg, the overall movement actions of an autonomous car in response to the wildly varied stimuli of an urban road environment) represent an enormous departure from the typical safety-critical assured function.

A particularly challenging source of uncertain behavioural requirements is human interaction. There is a need to specify how the AS should interact with humans, including coexistence, collaboration, handover of control. These are analogous to human-human interactions, and in our current practice most of those are severely underspecified.

Even if the needed functions can be precisely specified with high validity, the resulting complexity of those functions may present severe problems of verification.

Sub-challenges here include:

- How aspects of the AS development process will be assured, including, but not limited to:
  - Assuring potential hazards have been identified in a suitably rigorous manner;
  - Assuring the potential for risk transfer has been identified and appropriately controlled.
- How interactions between AS and their operators will be assured, including, but not limited to:
  - Managing potential safety argument implications of removing human senses as health monitoring functions;
  - Assuring the operator is in a position to take control (if required);
  - Assuring information is presented to the operator in an appropriate form;
  - Assuring the operator has a suitable understanding of the limitations and expected behaviour of the AS, including in off-nominal situations;
  - Assuring the AS is robust to the effects of communication delay and limited bandwidth;
  - Assuring that any systems that are intended to be advisory truly are advisory (ie, the operator can make an independent judgement and will not get vilified for doing so).
- How interactions between AS and second, and third, parties will be assured, including, but not limited to:

  - Assuring that all relevant second and third parties have been identified;
  - Assuring that the AS can communication with second and third parties (as required);
  - Assuring the AS can safely protect itself against hostile actors, both cyber and physical;
  - Assuring the AS is free from unintentional bias.
- How interactions between different ASs will be assured, including, but not limited to:
  - Assuring the AS can safely, and dynamically, form and reform connections with other AS with which it has been designed to inter-operate, including the implications of changing safety significance;
  - Assuring the AS can act safely within an environment that includes other AS, with which it has not been designed to inter-operate (and which may have been developed by competitor organisations).
- How the AS's behaviour in uncertain environments will be assured, including, but not limited to:
  - Assuring the AS can maintain safety in the absence of external services (eg, GNSS/GPS);
  - Assuring the AS can maintain safety in all relevant meteorological conditions and at all times of day or night;
  - Assuring the AS can adequately measure its environment and that the AS is robust to unexpected environmental measurements;
  - Assuring that an appropriate range of situations is covered during design, implementation and test activities.

6.2 Use of Novel Technologies

Extant proposals to implement the AS functions (eg, sense and avoid, object identification, communication of intent to peer vehicles, adapting to unexpected conditions, etc) frequently rely on technologies that have not yet been used in a safety-critical capacity before, such as neural networks, reinforcement learning, and visual objection identification. When juxtaposed with the extreme conservatism of current safety-critical software development standards ("you may not recurse, you may not dynamically allocate memory"), this presents a very serious challenge.

Sub-challenges here include:

- How the data used to develop (eg, train) the AS's behaviour will be assured, including, but not limited to:
  - Assuring that the TTV data is correct and suitably representative of operational data;
  - Assuring that independent validation data is available and that this covers: cases specifically designed to fool the AS; cases that may result from limitations in other parts of the AS (eg, sensor degradation, weather effects).
- How aspects of the AS development process will be assured, including, but not limited to:
  - Assuring requirements have been suitably specified, their intent has been maintained and they have been appropriately validated (including the implications of a lack of hierarchical decomposition);
  - Assuring typical errors have been avoided (ie, robustness testing);
  - Assuring supporting frameworks and libraries are suitable for their intended use;
  - Assuring that a suitable level of test coverage has been achieved.

6.3 Need for an Advanced Integrating Architecture

When a human operator is removed from a system and replaced by an autonomous controller, the system loses its central mediator subsystem; the subsystem that integrates and coordinates of the system's overall set of sensors, actuators and software functions. The autonomous system controller must therefore have a corresponding autonomous software architecture which performs this integrating function, and which does so in a way that can be assured.

This is particularly acute in the automotive space, where there is little tradition of vehicle-level software architecture design.

The nature of many interesting autonomous systems (again, including autonomous cars) is such that a great many of their apparently-distinct functions interact because they all influence a small number of high-risk physical properties (eg, the momentum of an autonomous car). The architecture therefore has the unenviable task of ensuring that all the system's functions interact only such that these properties remain safe. Its designers have the very unenviable task of assuring that it does so.

6.4 Need for a Lifecycle with Extensive Through-Life Design, Verification and Validation Activities

Autonomous systems may need an a lifecycle that weakens the traditional "development versus operation" split, initially because of the uncertainty around their novel functions and novel technology, then in the longer-term because of increasingly indirect specification of functions, for example, through learning or planning technology (as in "we can't specify this, but we can specify a learner which can").

Sub-challenges here include:

- How the AS will be assured during its use, including, but not limited to:
  - Assuring operational inputs are representative of the training and test data;
  - Assuring systems used to capture, transfer and analyse data related to monitoring of system behaviour;
  - Assuring software updates and system adaptations (including on-line learning) are appropriately managed, balancing frequency with assurance;
  - Assuring suitable monitoring is in place to detect AS-related errors (and invoke an appropriate system-level response).

This page is intentionally blank

Appendix A Acronyms

ADAS Advanced Driver Assistance System

AEB Automatic Emergency Braking

ANN Artificial Neural Network

AS Autonomous Systems

ATC Air Traffic Control

DSIWG Data Safety Initiative Working Group

GNSS Global Navigation Satellite System

GPS Global Positioning System

GTOLS GPS Take Off and Landing System

HAZOP Hazard and Operability Study

ML Machine Learning

NHTSA National Highway Traffic Safety Administration

NTSB National Transport Safety Board

SASWG Safety of Autonomous Systems Working Group

SCSC Safety Critical Systems Club

SSS Safety-critical Systems Symposium

TEVV Test, Evaluation, Verification and Validation

TTV Training, Test and Validation

UAS Unmanned Air System