Certification of Autonomous Systems

Robert Alexander, Martin Hall-May, Tim Kelly
Department of Computer Science, University of York
Heslington, York, YO10 5DD

Abstract

Many proposed autonomous systems are safety critical and will have to be certified before they can be operated. This paper reviews the challenges that autonomous systems pose for safety and the current state of relevant safety standards, concluding that autonomous system certification is possible under current regimes, but difficult given current safety analysis techniques. Some promising techniques are reviewed.

Keywords: safety, certification

Introduction

The SEAS DTC is concerned with developing technologies and methods for building autonomous systems (AS) which will operate in a range of military roles. Such systems clearly have the potential for life-threatening accidents.

A system capable of causing an accident that leads to human injury or death, or substantial material loss, is considered safety-critical, and before being deployed it must be certified as adequately safe according to applicable standards. The standards that apply vary with the type of system and the environment in which it will be operated. For example, the safety-critical systems procured and operated by the UK Ministry of Defence must now be certified against the requirements given by Def Stan 00-56 [1].

The need to certify autonomous systems is new, and consequently there is neither an established way of performing certification nor adequate advice on how this should be achieved. The specific technologies used in AS, and the complex environments in which they must perform, present further difficulties.

The next section uses the DTC vignettes to sketch some ways in which AS can be dangerous. This is followed by an exploration of why AS present problems for safety engineering. Relevant safety standards are then reviewed. The safety problems and certification requirements are drawn together to present some requirements for moving forward, and finally a selection of existing work is reviewed in the light of these.

Risks in the DTC Vignettes

Just as a number of capability challenges have been identified in relation to the vignettes, it is possible to identify a number of safety challenges. There are four main accident types that can occur in the ten vignettes:

1. Collision of autonomous vehicle (AV) with human pedestrian or vehicle with human occupant (or near miss, causing said vehicle to crash).
2. Human hit by AS combat capability.
3. Human exposed to threat due to AS inadequately or inaccurately reporting a threat.
4. AS action causes/triggers accident outside of its own capability.

Vignette Number	Vignette	Objective	AVs	Hazard	H. No.
7	Harbour Recon	Locate and recover cargo containers containing hazardous material in a flooded area.	USVs, UUVs	AV fails to recognise leaking container, exposes survivors to contamination.	4
8	Air Attack	Locate and neutralise a number of SAM sites in enemy territory before troops are called in.	UAVs	Misidentification of a civilian site as a SAM installation.	2
9	Urban Recon	Ensure house is clear (e.g. free of insurgents, booby traps etc.) and safe for troops to enter.	UGVs	Incorrect SLAM could mean that a room containing a threat is missed.	3
10	Route Maintenance	Patrol route, reporting threats such as mines, car bombs and snipers to command.	UGVs	AV travelling on wrong side of the road, while human-occupied vehicle is approaching.	1

Table 1: Example Hazards for Vignettes 7-10

While the first two hazard types (type 1 and 2 above) are direct consequences of the failure of a safety-critical function, type 3 can be caused by a failure to perform a safety-critical function at the system of systems level – this may be an example of poor reliability or availability leading to a safety problem. Meanwhile, type 4 is indicative of a hazard present in the environment, as opposed to one that an AS can cause on its own. For example, an AV runs over and inadvertently triggers a mine, or an AS gives a friendly position away to the enemy.

Table 1 shows some examples of the above-mentioned hazards in vignettes 7 through 10.

Problems for AS Safety

Complexity of the AS Environments

Autonomous systems must operate in complex environments. Fox and Das in [2] state *"Safety problems are difficult enough in 'closed' systems where the designers can be relatively confident of knowing all the parameters which can affect performance…"*, but that there are environments *"which cannot be comprehensively monitored or controlled, and in which unpredictable events will occur"* and that systems that have to operate in such environments *"may be exactly the kind of application where we want to deploy autonomous agents"*.

Jackson, in [3], notes that the much work in software engineering, for example, attempts to ignore the world and confine itself to analysis of software systems ('the machine'). He observes that this is prevalent in work dealing with the formal description and verification of software. This bias is clearly not sustainable for autonomous systems. Attempting to do this for autonomous vehicles, for example, would mean attempting to ignore the existence and importance of physical sensors, actuators and the external phenomena with which they interact. It is clear, however, that the behaviour, and hence the safety, of the vehicle will depend on these factors.

In order to evaluate the behaviour of AS under various conditions, models must be built of the environments that they will encounter. However, the correspondence of such models to the real environment must be evaluated, and this in itself is a difficult task.

Issues with AS Technologies

There are several classes of technologies used (or proposed for use) with AS that present novel challenges for safety certification:

First, there is the class of *model-based* systems, whereby the system makes decisions based on an explicit model of itself and the environment it occupies. This model embodies a large amount of explicit domain knowledge, and allows the autonomous system to predict the effects of its actions. The safe behaviour of the system depends both on the software that operates on the model (the 'engine') and on the model itself. There are parallels here with the simpler case of data-driven systems (see Storey and Faulkner in [4]) in that conventional techniques for safety analysis of software systems are not immediately applicable.

Extensions of those model-based systems are *model-building* systems that build their model over time. Because this model is built during operation it is not possible to validate the model ahead of time. It is therefore necessary to justify that the system will not build a model that will lead to it becoming dangerous.

While the model-building systems acquire data over time, the class of *learning* or *adaptive* systems attempt to extract explicit patterns or rules automatically from that data. Cukic, in [5], observes that the functional properties of an adaptive system cannot be inferred by a static analysis of the software. Kurd, in [6], identifies key challenges as being the difficulty of understanding the model that the system has learned (behaviour transparency and representation), preventing violation of identified safety requirements (behaviour control) and managing the trade-offs between safety and performance.

The effective exploitation of system or world models requires the use of *planning* techniques, whereby the system searches for possible paths through the states of the model that will allow it to achieve its goals. Brat and Jonsson observe that *"Verifying a planner is an enormous challenge considering that planners are meant to find intricate solutions in very large state spaces"* [7]. It is therefore difficult to show that a given planning system will behave safely in all combinations of models and situations.

Many of the technologies proposed for use in AS provide *probabilistic functions*, in that the complexity of their interaction with their environment is such that their behaviour under any given circumstance can only be described probabilistically. Hawkins, in [8], notes the difficulty of developing probabilistic systems with behaviour predictable enough to be used in a safety-critical role, given the very low probabilities of hazardous failure that are required. Probabilistic functions can be subjected to statistical testing, but it is acknowledged in the software safety community that such testing cannot give a satisfactory level of safety assurance on its own; McDermid and Kelly note, in [9], that at best statistical testing can show *"a failure rate of about 10^{-3} to 10^{-4}"*.

Certification Context

Blanquart et al, in [10], provide a brief survey of software safety standards and assess their applicability to autonomous systems.

It can be noted that these standards (at least in the versions extant at the time of Blanquart's survey) are largely prescriptive and process-based. As such, they

recommend a set of techniques and methods for safe development of software, but Blanquart et al note that these standards *"pay little attention to autonomy and to the particular advanced software technologies for system autonomy"*, and that *"In practice the recommended set of techniques and methods for safety-related software may not be easily applicable considering, e.g., the size and complexity of the software and of the input and state domains, the dependency of the software behaviour on knowledge bases, etc."*

Since the Blanquart survey was published (in 2004), the UK Ministry of Defence has issued a new general safety standard (Def Stan 00-56 Issue 3) that has the potential to make it easier to certify novel classes of system. In addition, a number of UAV-specific standards have been proposed.

Def Stan 00-56 Issue 3

Def Stan 00-56 Issue 3, "Safety Management Requirements for Defence Systems" [1], published in December 2004, presents a possible path towards a certification solution. Rather than prescribing a development process and a set of techniques, which may not be applicable to novel types of system, it allows the developer of a system to justify its safety using a safety case structured to present a risk-based argument that the system is safe.

This is a 'product-based' safety argument approach rather than a 'process-based' one; it involves the presentation of evidence that the actual developed system is safe, as opposed to merely showing that it was developed using accepted good practice. This gives good scope for the certification of novel classes of systems, such as AS; the system can be certified if a compelling safety case can be built for it.

For military applications, 00-56 Issue 3 is particularly significant because **all** new acquisitions by the UK Ministry of Defence must have a safety case presented in line with this standard.

00-56 has a strong emphasis on the provision of analytical evidence, as distinct from test or demonstration evidence (or 'qualitative' evidence such as the use of a good process). The actual text from the standard is: *"Within the Safety Case, the Contractor shall provide compelling evidence that safety requirements have been met. Where possible, objective, analytical evidence shall be provided"*. Justification for this position, and indeed for the approach taken by 00-56 overall, can be found in [11].

There are some problems with the use of 00-56 as it stands. First, it states that the developer of a system must systematically determine, for each identified risk, the severity of the consequence and the likelihood of occurrence. However, as noted in the introduction, the main motive of the use of autonomous systems is for those situations where the full details of the operating environment cannot be known ahead of time. It could therefore be difficult to carry out risk estimation as required by 00-56 using conventional techniques.

A second problem is that although 00-56 provides a framework in which the safety of any system can potentially be argued, there is no extant guidance on how to do this for AS. There is therefore a need for methods and patterns to be developed for producing safety cases given the challenging technologies, environments and tasks of autonomous systems.

UAV-specific Standards

Several UAV-specific standards and guidance documents have recently been issued. Given space limitations, we will consider only two: CAP 722 [12], a document issued by the Civil Aviation Authority (CAA) providing guidance on operating Unmanned Aerial Vehicles in UK airspace, and Def Stan 00-970 Issue 4

part 9 [13], which gives *"Design and Airworthiness Requirements"* applicable to UAVs procured by the MoD.

Generally, the extant UAV standards are very conservative in terms of level of autonomy. For example, 00-970 requires that the UAV operate using a pre-planned flight path which is uploaded to the UAV and which can be changed (by the operator) at any time during flight, and also states that *"direct, online control of the UAV flight path shall be avoided where possible"*. This is much less autonomy than the DTC vignettes, for example, include.

CAP 722 proposes that UAVs should achieve *equivalence* to manned aircraft – the technologies used by the UAV must be demonstrably equivalent to human capabilities. For example, sense-and-avoid must provide the same level of collision avoidance as see-and-avoid. Furthermore, it proposes that UAVs should provide *transparency* – the Air Traffic Control Operator (ATCO) must not have to apply a different set of rules or assumptions when providing an Air Traffic Service to a UAV. It follows that the CAA want to avoid changes to the existing Rules of the Air. It is not clear how this restriction to human equivalence is to be achieved, and in any case this approach may sacrifice valuable opportunities for achieving increased levels of safety.

The Way Forward

The preceding sections have explored the problems posed by AS environments and technologies, and by current certification regimes. It can be seen that there is potential for certifying (at least military) AS using 00-56 compliant safety cases. This, however, will require:

- Solutions to the problems with the identified AS technologies.
- Safety analysis techniques that can derive the effects of complex environments.
- Ways to achieve (and argue) coverage of all risks in a complex AS.
- Means of deriving and presenting *analytical* evidence for inclusion in the safety case.

Relevant Technologies and Methods

There are a substantial number of attempts to tackle the problem of safety in AS, and these need to be reviewed against the requirements identified in this paper. This is necessarily a very brief survey – further details and examples can be found in the report produced by the authors [19].

Safety-Critical Artificial Neural Networks

Kurd, in [6] discusses the use of Artificial Neural Networks (ANNs) in safety-critical applications. An ANN is an example of an adaptive system, and therefore presents a variety of problems for safety certification.

Kurd describes an ANN architecture that provides a human-readable and comprehensible representation of the rules it embodies (in contrast to the 'black box' nature of conventional ANNs), and allows individually meaningful rules to be extracted and inserted. It therefore makes it possible to control the behaviour of the ANN. Kurd provides a method for deriving safety requirements for ANNs, and the ability to observe and control the network allows these to be imposed. He also provides guidance on building a safety case, which shows how the safety of an ANN implementing these safety properties can be argued effectively, using analytical evidence of the system's safe behaviour.

The work is a strong general example of what is needed to allow a novel technology to be used in a safety-critical system. ANNs, however, are only one example of an adaptive technology. Comparable work will be needed for other techniques.

Formal Analysis using Kripke Modelling

The team at Cranfield University (Defence Academy Shrivenham) present the results of a series of feasibility studies focused on a formal approach to modelling the interaction of autonomous vehicles. In their work [14] an intuitive, yet mathematically rigorous, approach of Temporal Logic and Kripke modelling is presented for representing a co-operative, decentralised group of autonomous vehicles moving under the conditions of environmental uncertainty.

The Temporal Logic based approach has been successfully used to design and validate zero-fault tolerant systems such as hardware chips and avionics software, outperforming traditional methods like inspection, testing and simulation, and axiomatic (theorem proving) approaches to program verification.

In the feasibility studies the scenarios entailed prototypes of a fundamental task required for a group of autonomous vehicles. This task is that of coordinated arrival on target, despite different launch points, communication disruption and presence of unknown obstacles.

The feasibility studies have demonstrated the natural ability of the approach to scale up because it allows the behaviour of individual entities to be abstracted into descriptions of overall system states. The output of the approach is analytical, and hence highly suitable for use in a Def Stan 00-56 safety case.

Formal Analysis using Soar, CSP and Model Checking

QinetiQ have developed an approach using formal mathematical assessment techniques to verify properties of autonomous agent systems. Descriptions of agent logic in the Soar artificial intelligence language [15] are automatically translated into the Communicating Sequential Processes (CSP) process algebra [16]. The CSP representation can be analysed by the FDR2 model checker to verify that the system implements desired properties. When the implementation satisfies the properties that have been specified for it, the CSP is converted to Handel-C which can be implemented directly in hardware.

The approach potentially allows for the creation of complex, deliberative agents (using the expressive power of the SOAR language) and for the representation of complex agent environments (modelled in CSP).

The approach is strong in that it starts from a highly expressive language designed for human comprehension and creation and translates it into a form that is amenable to formal analysis and from there generates a representation that can be compiled directly to hardware. Its output is analytical evidence and hence valuable under a 00-56 regime.

HIRTS DARP

Strand 2 of the HIRTS DARP project focussed on the safety and dependability of Systems of Systems (SoS). Although the work was not restricted to autonomous systems (it also included the actions of human-operated systems) it is clearly applicable to interacting groups of AVs.

In [17], Alexander describes an approach to the hazard analysis of SoS using simulation models. Hazards are identified by running the model with a wide variety of anticipated deviations, and using machine learning to extract patterns from the results; this avoids some of the problems with traditional statistical analysis.

Hall-May, in [18] shows how a ‘safety policy’ can be defined to ensure the safety of SoS, by imposing obligations and restrictions on the behaviour of the system’s constituent entities. The derivation of safety policy can be based on

prior hazard analysis, or performed directly from agent models

The common use of simulations for autonomous system prototyping means that they are highly amenable to hazard analysis through simulation. Simulation also provides a vector for the description of complex environments and investigation of their effects on the system. Hall-May's work on safety policy is then applicable to ensure the safe interaction of multiple systems, and the goal structure representation offers a great advantage over traditional free text policies in that reasoning and justification behind each policy rule is clearly expressed.

Conclusions

It is clear that proposed autonomous system technologies, environments and applications present problems for safety analysis and safety assurance, and therefore for certification. These problems give rise to requirements for safety research.

There is published work on this topic, but there is nothing that provides safety assurance adequate for certification, or a safety analysis process that can show, to an adequate level of confidence, that a given autonomous system is adequately safe. There are no safety standards extant for non-UAV AS, and much of the UAV-specific standards work has a (questionable) emphasis on achieving human equivalence rather than optimum safety.

There is existing work on safety analysis of AS, but much of it is point solutions which are only applicable to a single technology or to components of an overall autonomous system. Further development of this work is needed.

Defence Standard 00-56, in its latest form, has been abstracted to a fundamental set of safety objectives that can be applied to many classes of systems. However, there remain significant difficulties in realising these objectives where conventional analysis techniques and forms of safety argument cannot be applied to AS and their underlying technologies. There is therefore a strong need for definition of a general AS safety lifecycle, expansion and development of existing safety analysis methods, and for substantial guidance on the development of 00-56 compliant safety cases.

The authors have produced a much larger report [19] which expands on all the material presented in this paper.

References

[1] MoD Interim Defence Standard 00-56 Issue 3 - *Safety Management Requirements for Defence Systems*, December 2004.

[2] Fox J and Das S, *Safe and Sound: Artificial Intelligence in Hazardous Applications*, 2000, MIT Press.

[3] Jackson M, *The World and the Machine*, 1995, Proceedings of the 17th international conference on Software engineering.

[4] N Storey and A Faulkner. *Data - The Forgotten System Component?* Journal of System Safety, 39(4), 10-14, 2003.

[5] Cukic B. *The Need for Verification and Validation Techniques for Adaptive Control System.* In Proceedings of the Fifth International Symposium on Autonomous Decentralized Systems (ISADS 2001), pages 297–298. March 2001.

[6] Kurd Z. *Artificial Neural Networks in Safety Critical Applications.* Ph.D. thesis, Department of Computer Science, University of York, 2006.

[7] Brat G and Jonsson A. *Challenges in verification and validation of autonomous systems for space exploration*. Proceedings of IJCNN'05:Performance of Neuro-Adaptive and Learning Systems: Assessment, Monitoring, and Validation. 2005.

[8] Hawkins R and McDermid J. *The use of Bayesian Networks in Critical Applications.* In Proceedings of the 23rd International Systems Safety Conference (ISSC 2005). 2005.

[9] McDermid J A and Kelly T P. *Software in Safety Critical Systems: Achievement and*

Prediction. Nuclear Future, 2(3):140–146, May 2006.

[10] Blanquart J P, Fleury S, Hernek M, Honvault C, Ingrand F, Poncet J C, Powell D, Strady-Lécubin N, and Thévenod P. *Software Product Assurance for Autonomy On-Board Spacecraft*. Proceedings of DASIA 2003 (ESA SP-532), pages 69A–69G. June 2003.

[11] Caseley P R, Tudor N, and O'Halloran C. *The Case for an Evidence Based Approach to Software Certification*. Unpublished Dstl / MoD / QinetiQ report, 2003.

[12] UK Civil Aviation Authority. CAP 722— *Unmanned Aerial Vehicle Operations in UK Airspace: Guidance*. The Stationery Office, November 2004.

[13] Defence Standard 00-970 *Design and Airworthiness Requirements for Service Aircraft* Issue 4. Part 9 — UAV Systems. Ministry of Defence, January 2006.

[14] S. Jeyaraman, A. Tsourdos, R. Żbikowski, B.A. White, *Kripke Modelling Approaches of a Multiple Robots System with Minimalist Communication: A Formal Approach of Choice 2006,* International Journal of Systems Science Vol. 37, No. 6, pp.339–349

[15] Lehman J F, Laird J, and Rosenbloom P. A *Gentle Introduction to Soar, an Architecture for Human Cognition*: 2006 Update. http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/GentleIntroduction-2006.pdf, January 2006.

[16] Hoare C A R. *Communicating Sequential Processes*. Prentice-Hall International Series in Computer Science. Prentice-Hall, 1985.

[17] Alexander R, Kazakov D, and Kelly T. *System of Systems Hazard Analysis Using Simulation and Machine Learning*. Proceedings of the 25th International Conference on Computer Safety, Reliability and Security (SAFECOMP '06), pages 1–14. September 2006.

[18] Hall-May M and Kelly T P. *Using Agent-based Modelling Approaches to Support the Development of Safety Policy for Systems of Systems*. Proceedings of the 25th International Conference on Computer Safety, Reliability and Security (SAFECOMP '06), pages 330–343. September 2006.

[19] Alexander R, Hall-May M and Kelly T P, *Certification of Autonomous Systems*, January 2007, SEAS DTC.

Acknowledgements

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence. The authors would like acknowledge the support of BAE Systems (Andy Cox, Tim Doggart, Jane Fenn, Richard Hawkins, Brian Jepson, Andrew Miller, John Shuttleworth, Malcolm Touchin), QinetiQ (Simon Evans, Chris Greenfield, Richard Harrison, Colin O'Halloran, Philip Vale) and Cranfield University (Brian White, Antonios Tsourdos, Rafał Żbikowski).