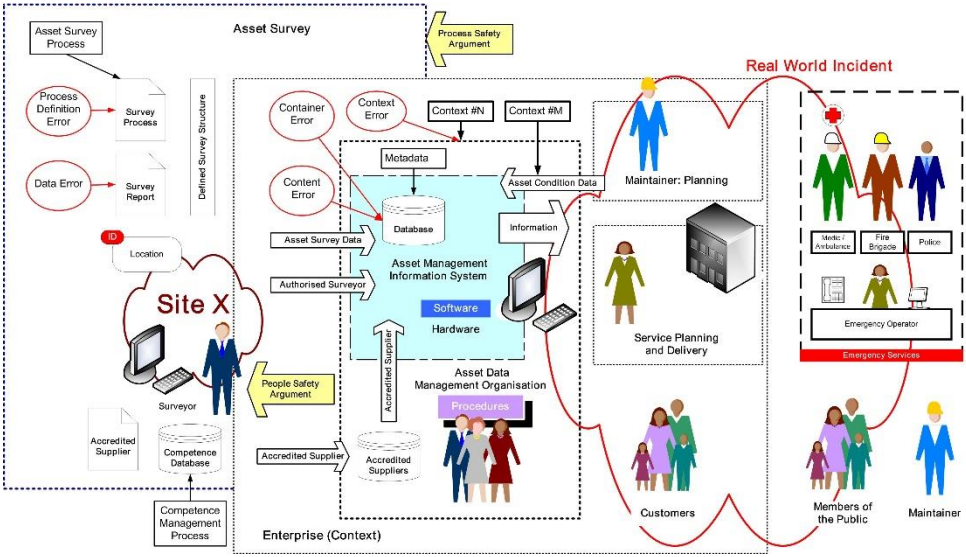
Data, Data Everywhere ...



The easy availability of data has led to Data-Centric Systems (DCS) that are highly data-defined and data-driven. The nature of how the data is used, and depended on, is also radically changing with the advent of Machine Learning (ML) and Autonomous Agents (AA). Alastair Faulkner discusses the challenges of assuring Data-Centric Systems.

Data (in all its forms) is often unchallengeable, unverifiable, ubiquitous, unrecorded and invisible. Yet this data increasingly determines the behaviour of systems and through this behaviour our access to products (goods and services). Data may be internal or fed to systems with safety responsibility. As a result, data error or omission may go undetected with potentially hazardous or catastrophic consequences. There may also be consequent damage to assets. Failure of such systems may also contribute to harm indirectly through incorrect decisions made by actors (human or computer) who rely on, or trust, these systems and the data they supply. How should we reason about Data-Centric Systems (DCS) so that our reliance on, or trust in their correct operation can be justified?

The domain of safety-related DCS is immature, and as such, the safety community has yet to reach consensus on many aspects of architecture, design, implementation, operation and maintenance. Past debates over software best practice and guidelines serve to illustrate the difficulties in reaching consensus. The questions raised in this article require further consideration and I encourage the reader to participate in the debate.

"Swimming in sensors, drowning in data"

The widespread application of infrastructural technologies breakdown old barriers creating an age of connected systems. Seemingly innocuous sensors use highly capable computational platforms. These technologies have the potential to produce vast quantities of data. [1] This combination of production, communication and consumption shifts the focus away from hardware and software to data. This leads to self re-enforcing pressures to create evermore data reliant systems.

"Data, once scarce, is now superabundant"

Superabundance [2] recognises that our ability to produce data exceeds the resources available to transform it into information. This simple statement creates additional challenges beyond production and consumption. Which data should be retained (and why)? For how long? Who should have access? More importantly, what can data be relied on, and why is it good enough?

Before we explore these issues, it is worth a small thought experiment. Consider a defined system. A suitable and sufficient safety analysis identifies several safety functions and associated safety requirements. The apportionment, based on the system architecture, determines at least one of these safety functions is higher than SIL2. A solution constraint requires the use of only hardware and software with the limited use of data. A dual-channel 2oo2 architecture is selected using diverse implementation. This conventional approach fits within the confines of many safety standards and existing safety solutions.

At the last minute, an alternative solution is proposed. It consists of a database supported by generic hardware and software. It is claimed that failures of the generic hardware and software have no (safety) impact. That all the behaviour is described by and contained within the database. Therefore, the safety functions are implemented using data. The existing safety standards constrain safety assessment. For example, two questions arise; firstly, where is the required diversity for this SIL2 system; and secondly, how can this solution implement a dual channel 2oo2 architecture?

"High integrity safety systems require strong contexts and depend on strong-data"

An inescapable conclusion

The debate as to whether data is a separate system component is over. It requires an equitable treatment of data and raises awkward questions as to the safety of existing data reliant safety systems. Nevertheless, the case for data builds day by day; it is now inescapable.

Data Safety Challenges

The combination of data volume and its use to characterise, parametrise, configure and describe the system behaviour provide an overwhelming argument. Data **is** a separate system component and often the dominant element. Data has always been present, typically though not exclusively in lower volumes where its influence is constrained to specific functional areas and domains. Infrastructural technologies are the primary catalyst. It is essential to create foundations as constructs for data safety.

Data is only *valid* in a defined set of contexts

Context is a difficult concept to express. A context may be closed and separate, or open interacting with other different types and categories of systems and operating environments. Its definition includes any information used to characterise the situation of an entity [3]. An entity is a person, place, or object that is considered relevant to the interaction between an actor, an application, systems and their environment, including the actors and systems element themselves [4]. Therefore, defining a context concerns capturing the conceptual structures and frameworks used to construct the system, its boundaries, and its utility in the operational environment. Typically, complex systems contain many different types of actor. Each type of actor has a viewpoint that contains a subset of elements of the context [5]. Extensive datasets are often associated with data ecosystems that include collections of infrastructure, analytics and applications used to capture and analyse data [6].

The properties of a data context relate to its antecedence. For example, strong-data is finite in volume, very specific, such as a sensor or medical record [7]. In contrast, one source of weak-data is data derived from sources often vast in quantity, typically fuzzy and ambiguous, by data analytics [7]. High integrity safety systems require strong contexts and depend on strong-data.

Data Definition

To support safety analysis, data should be defined. The data container may be structured, unstructured or semi-structured and represented in one or more data models. This analysis leads to consideration of the role of data and the reliance on the correctness of the data. Simple data may be a scalar, an array or a table. Data's role is not limited to characterisation, parameterisation and configuration. Data is also the basis of the definition and provision of function, flow and service. In high integrity safety systems, EN 61508 requires diversity as a different means of performing a required function [8]. Diversity is an architectural approach to addressing the issue of Single Point of Failure (SPoF) or Common Cause Failures (CCF) leading to a safety event. Standards typically identify two types of diversity: mechanistic and conceptual. One of the many challenges for Data Safety is the development of a consensus on data diversity.

Data Content

Much literature addresses data quality as a measure of the *content* stored in one or more *containers*. The IEC 25000 [9] series of standards provides a complete treatment of data quality. Data Safety cannot be assured using data quality measures alone.

Bottom-up versus Top-down

Real-world data plays an essential role in all safety systems and their management. As existing data-centric systems with safety responsibility are identified, post-implementation safety analysis is required. Experiential (bottom-up) techniques are necessary for the identification of hazards, their possible removal, mitigation and management. These are the core aspects of all Safety Management Systems (SMS). Experience Based Quantification (EBQ) is a collection of bottom-up techniques used to justify decisions based on data collected from the real world. Competencies derived from EBQ are often confined to those

managing operational process rather than informing and enriching corporate memory. Dis-associations induced by EBQ create a disconnect between corporate management and therefore reduce their ability to implement appropriate strategies.

Once system safety responsibilities are identified, the implementation of safety requirements and any remedial actions may prove difficult. For example, an examination of data errors in a criminal justice system database is only one part of the process, without the resources to correct those errors, they remain in place [10]. Efforts to correct existing data quality issues are bottom-up.

In contrast, top-down safety management follows a more traditional path from requirement, design, implementation, verification, installation operation and maintenance. This implies the use of development and safety lifecycle. On completion, the system is installed, and any changes are instantiated at baseline release. Inferred in this mechanism is that all hazards are known at the time of implementation.

The influence of data on the system extends beyond individual data elements or collections. One means of managing the safety risks associated with data require the consideration of context, container and content. Hawkins et al. [11] have developed 4Plus1 safety assurance principles for software as a pragmatic approach that considers both top-down and bottom-up. The 4Plus1 safety principles are reinterpreted by the author for Data Safety in Table 1.

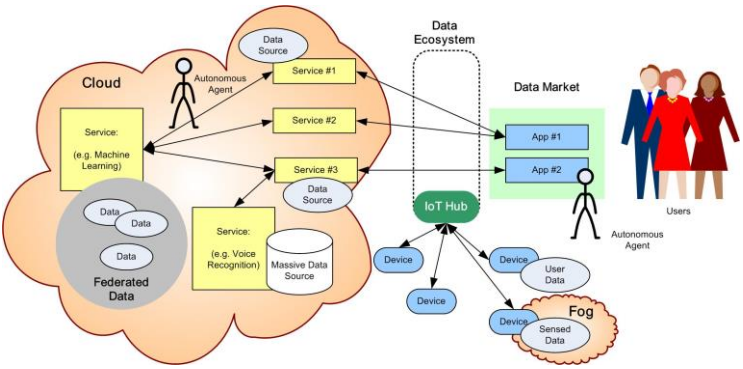
Table 1: 4Plus1 Data Safety Principles

Principle	Description
1	(Top Down) (Data) safety requirements shall be defined to address the data contribution to system hazards and associated risks.
2	(Top Down) The intent of (data) safety requirements shall be maintained throughout requirements decomposition (and apportionment to data components).
3	(Top Down) (Data) safety requirements shall be satisfied.
4	(Bottom Up) Hazardous behaviour has been identified and mitigated.
Plus1	The confidence established in addressing the (data) safety principles shall be commensurate to the contribution of data to system hazards and associated risks.

As the lifecycle progresses, the requirements and design are progressively decomposed. In this way, safety requirements relating to action response requirements are decomposed into knowledge requirements, then information requirements and finally, data requirements. A pragmatic approach requires consideration of the technological constraints, including assessing the requirements placed on the data elements such as sensors and their acceptable Operational Design Domain (ODD). A more detailed design is created. The intent of the requirements must be maintained as the safety requirements are decomposed. Data safety requirements must ensure that the safety intent is maintained as the data architecture and individual data elements emerge.

Data Safety Architectures, Design, Techniques and Measures

It is relatively easy to define an acceptable ODD for a sensor. This task becomes more complex when combing multiple sensors in an array. Sensor fusion addresses how different sensors are combined. Sensors are only one source of data.



Architecture

How should all these elements be combined into an appropriate set of safety architectures that provide robust defences against SPoF and CCF? This consideration illustrates the requirement for multi-channel systems and diversity. In the absence of safety community consensus, defining acceptable ODD for safety subsystems and systems will be challenging. Many of these systems will not be stand-alone but operate and co-operate with others in a hierarchy.

Design

Data Safety spans the spectrum of implementation from high-integrity protection systems to information systems. The increasing use of agile development creates challenges for safety analysis and safety acceptance due to the short iterative cycles and the currency of the documentation. Data-defined and data-directed systems often use generic hardware and software, their form and behaviours are wholly described by data.

As systems become more extensive in scale, scope and complexity, a system will likely consist of multiple instantiations. This will be the case for Autonomous Vehicles (AV). What design elements are required for version, configuration management to provide for maintenance, upgrade and incident investigation? One suggested method for addressing this is that an acceptable AV ODD should also include interaction with other AVs and their environment.

Techniques and Measures

A safety community consensus for the recommendation of techniques and measures [12] is essential to the process of assurance and safety acceptance. The rigour of their use provides evidence and sets the competence, training and experience requirement on the developers. This becomes a circular argument, as, without safety community consensus for data architecture, design and implementation of each system safety assessment and acceptance are highly dependent on the individual assessor. Therefore, one option is to include a risk of 'failure to gain acceptance' in the project risk register.

Data Consumption

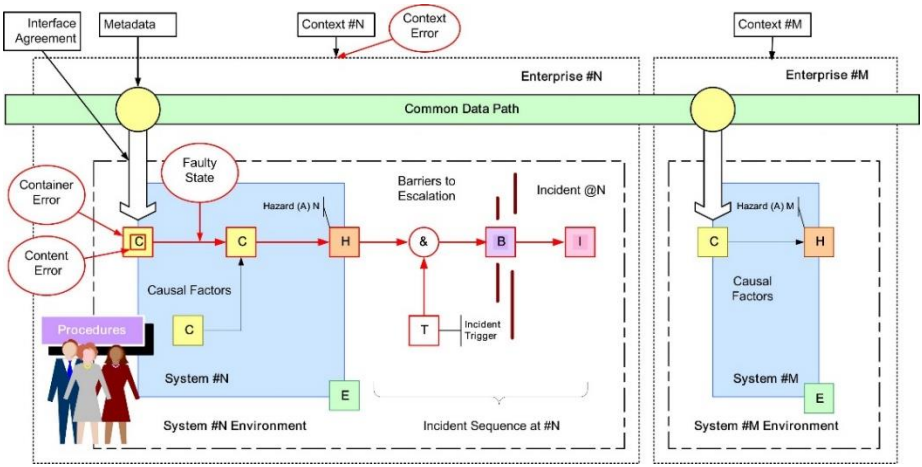
With a reliance on data, data integrity requirements are essential.

Given the contribution of data errors to the system behaviour safety analysis is likely to result in the apportionment of a high percentage of the safety risk to the data component. Data-Centric Systems (DCS) are unlikely to be stand-alone, their data integrity requirements are to be satisfied at each interface (and by implication the data path (data supply chain)).

Data Sources and Production

The ready availability of some data does not qualify it for use in a safety system. Solid foundations are essential to all engineered systems. This is also true for DCS. Data may originate from many sources; it may be highly processed and transformed. Its ownership may change many times. Who then will be liable for data error and consequent losses?

Is it reasonable to require an acceptable ODD for each element of the data path, or should the ODD be limited to the point of delivery (interface) of the consuming system? It depends on the use made of the data and the influence of data errors on the consuming system. A single data path might supply multiple systems. Only one of those systems might fail due to a specific data error.

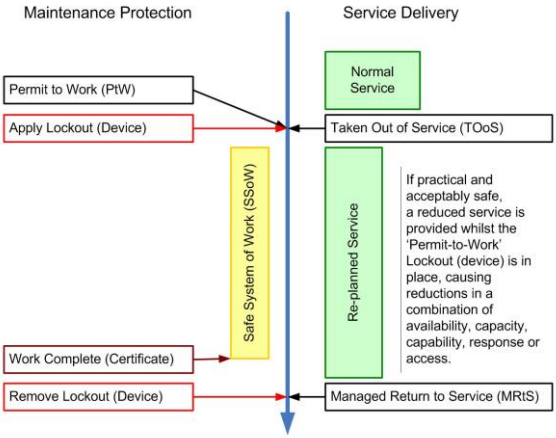


A DCS may require extensive data. Not all of this data will be needed by all the types of actors using the DCS. Classic safety engineering requires the identification of one or more boundaries, typically, although not exclusively interfaces. In this classical world, errors, faults and failures give rise to hazards at the system or component boundary. Where are these boundaries in extensive data? One suggested method for addressing this is to develop the concept of Interface Agreements (IA) as a means to describe both physical and virtual boundaries.

Interface Agreements

An interface is a point where two systems, system and environment, subjects, organisations, etc. meet and interact [13]. Conmy [14, 15] proposes Interface Contracts as a set of one or more [safety] constraints which must be upheld. These are typically used to describe high-integrity safety rely-guarantee (theorem) constraints. In a more generalised form, IA [16] are a set of one or more constraints which must be upheld by system components to meet one or more requirement(s).

Critical Control Points



Creating and installing a system also requires consideration of its maintenance. How should a system element be safely taken out of service? One suggested method develops the concept of a Critical Control Point: (CCP) as an IA used to provide data on the status, performance and behaviour of the system across the operational context. A CCP provides one means to implement Permit-to-Work (PtW).

Typically, a PtW is a management procedure where only persons with specific authority will sign a permit on which ostensibly the life of a worker might depend. To this end, responsibility for the PtW rests with the person in charge of the operation for which the permit is required [17].

Safety Acceptance

If safety acceptance to a safety standard such as IEC 61508, then, how should a DCS be assessed for compliance? Taking the assessors point of view, the overall architecture and hardware and software components are well-defined. The assessor is constrained by the standard, which says very little about data. What should the assessor do when a majority of the safety integrity requirements are apportioned to the data component?

Extensive scale, scope and complexity requires the assessor to recognise that elements of the system may be in different states at different times. It is proposed that acceptance and approval should consider:

- Staged start-ups
- Staggered start-ups
- Asynchronous start-ups (large distributed systems)
- Synchronous start-ups (aligning timing requirements and hand over [live to hot stand by])
- Localisation, Segregation and Quarantine (treatment – inoculation)
- Maintenance requirements and restrictions
- Operation without a Safe System State

Operation and Maintenance

Safety is a property of the operational system [18]. Where data describe the system and its behaviours, we should not assume that a system change will use the change procedures normally associated with the development lifecycle, change management, versions and baselines. Shortly after the widespread introduction of AVs, it is easy to imagine many vendors, each with many models and versions operating with conventional vehicles. In this AV system, there will be many contexts, containers and multiple (possibly duplicate) content.

We should not assume that the user is human. An actor may be an individual, entity, or combination of product, people and process. The role of the actor will increasingly be undertaken by one or more AAs. Where an AA is an entity operating on the owner's behalf as an actor without interference from the ownership entity. Typically, these are products that incorporate varying degrees of Machine Learning (ML) [19, 20].

It is proposed that the use of actors requires the definition and enforcement of identity management, such as the consideration of:

- Authentication
- Authorities
- Fraud Detection, Enforcement and Management
- (False) Identity [Agents, Users, Customers] (Joiners, Leavers and Renewal)
- (False) Identity (Components, Combinations of Components, and Systems)
- All identities (should) expire and need to be renewed? (over what period)
- Attacks and Malicious Event Response

Incident Investigation

Over the last five years, we have been fortunate to live through a period of relatively few major incident and accidents. One consequence has been a complacency and reduced focus on safety management activities. Many senior and experienced practitioners have reached or are approaching retirement.

In the absence of major incidents, existing safety margins in existing systems have proved sufficient. These technologies are now mid-life and approaching obsolescence; elements of these systems now require replacement. Fit-form-function replacements may be based on DCS.

Data is a decisive and disruptive technology; therefore, in the absence of a definition of safety-related DCS architectures or other desirable properties, the incident investigation method needs to be tailored to each context. Also, the use of data ecosystems requires characterisation for its potential influence on the incident. This characterisation is the basis of the formalised and documented approach to its investigation.

Data Safety

With the emergence of data as a separate system component arises the requirement for the consideration of Data Safety. In this sense, all that is argued for is equality with the other safety system components. The development of a safety community consensus on architecture, data structures, techniques and measures.

Data will enable and facilitate dynamic behaviours. This new dynamism finds form where data defines the system and its behaviours. Potentially, the system has escaped the controls normally associated with the development environment. In this sense, deployed products are 'unfinished'; their behaviours are modified by AA and ML through their operational experience. Data-Centric Incidents (DCI) will be caused by data component errors. Data ownership will become a contentious issue. Who will be liable for errors associated with data?

Conclusions

Hopefully, this article has provided some insight into the safety management challenges associated with DCS. One of the most profound changes is the shift from development based approaches to the creation of safety products to the operational arena. The ease with which data can be changed provides an irresistible economic factor for the proliferation of DCS.

As safety practitioners, we address engineered systems, created by a designer, to implement a design intent, developed, operated and maintained to that design intent. We commonly consider products as engineered artefacts. As a divisive technology, data will induce a paradigm shift. The underlying message of this article is the evolution of known and mature systems safety concepts, rather than revolution. DCS are only one aspect of change. Those organisations that use DCS will become increasingly reliant on them and in doing so become Data-Centric Organisations (DCO). One aspect of this evolution is to address increasingly open systems with external interactions across the system boundary. Increasingly open systems extend existing cyber-security risk and highlight issues of identity. It is proposed that that identity will be extended to ensure the unique labelling of attributes of the object (system resource) being accessed and of the actor requesting access in a given context.

In developing this article, it follows that there will be DCIs. How should they be investigated? Unlike physical incidents, DCIs may not leave physical witness marks such as tyre skid marks. Data error or omission may go undetected, and may also contribute to harm indirectly through incorrect decisions made by actors (human or computer) who rely on, or trust, these systems and the data they supply. Without proper design, DCS and DCO may become interconnected, interdependent, and as a result, will not be analysable.

Finally, the proposition that 'data is a separate system component' has widespread repercussions which are not simply limited to a system being comprised of hardware, software, actors, process and data. Data quickly becomes the dominant component. Without safety community consensus addressing context, container and content, it is unclear how the safety risks associated with data will be adequately managed.

References

[1] Lt. Gen. David A. Deptula. Military 'Swimming In Sensors and Drowning in Data'. URL: http://www.nationaldefensemagazine.org/articles/2009/12/31/2010january-military-swimming-in-sensors-and-drowning-in-data

[2] Kenneth Cukier. Data, Data Everywhere. The Economist, 2010. URL: https://www.economist.com/special-report/2010/02/27/data-data-everywhere

[3] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the CHI 2000 Conf. on Human Factors in Computer Systems, 2000

[4] Robert Flood and Ewart Carson. Dealing with complexity: An Introduction to the Theory and Application of Systems Science. Volume 2nd ed. ISBN 978-0306442995. Plenum Press, New York, NY, USA, 1993

[5] BKCASE Governance and Editorial Board. Engineered System Context. 2017. URL: https://www.sebokwiki.org/wiki/Engineered_System_Context

[6] Marcelo Iury S. Oliveira and Bernadette Farias Lóscio. What is a Data Ecosystem? 978-1-4503-6526-0. ACM, 2018, 74:1–74:9

[7] Shomit Ghose. Engineered Influence: Weak Data, Machine Learning and Behavioral Economics. 2017 Sutardja Center for Entrepreneurship and Technology's annual journal AIR (Applied Innovation Review), 2017. URL: https://scet.berkeley.edu/engineered-influence-weak-data-machine-learning-behavioral-economics/

[8] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviations. International Electrotechnical Commission, 2010

[9] IEC 25000: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. International Electrotechnical Commission, 2014. URL:http://iso25000.com

[10] Bill Blackburn, Paul Hampton and Mike Parsons, Data in Police and Criminal Justice Systems: How data errors could lead to harm to innocent citizens (or, how the film "Brazil" was spot-on), in Evolution of System Safety, Proceedings of the Twenty-sixth Safety-critical Systems Symposium, York, UK, 6th-8th February 2018, https://scsc.uk/SCSC-140

[11] Richard Hawkins, Ibrahim Habli, and Tim Kelly. The Principles of Software Safety Assurance. International System Safety Conference (ISSC), Boston, 2013

[12] Data Safety Guidance. Version 3.1. ISBN-13: 9781793375766. Safety Critical Systems Club - Data Safety Initiative Working Group, Kindle Direct Publishing Platform, 2019, https://scsc.uk/SCSC-127D

[13] Interface - Definition. Oxford Living Dictionaries. URL: https://en.oxforddictionaries.com/definition/interface

[14] Phillipa Conmy, John McDermid, and Mark Nicholson. Safety assurance contracts for integrated modular avionics. Volume 33. SCS '03 Proceedings of the 8th Australian workshop on Safety critical systems and software, 2003, pages 69–78

[15] Phillipa Conmy. Safety Analysis of Computer Resource Management Software. University of York, PhD Thesis, 2005

[16] Alastair Faulkner and Mark Nicholson, Data-Centric Safety - Challenges, Approaches, and Incident Investigation, Elsevier, 2020, Version, ISBN 978-0-12-820790-1

[17] Institution of Engineering and Technology (IET). Safe Systems of Work (Health and Safety Briefing No. 32). 2015

[18] Alastair Faulkner: "Safety Arguments for Use with Data-driven Safety Systems", Proceedings of the fourteenth Safety-critical Systems Symposium, pp 263-276 ISBN: 1-84628-333-7, Bristol, UK 2006.

[19] Stan Franklin and Art Graesser. Is it an Agent, or just a Program?:A Taxonomy for Autonomous Agents. Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996. URL: http://robotics.cs.tamu.edu/dshell/cs631/papers/franklingraesser96agents.pdf

[20] Jae-Gil Lee et al. "Can Autonomous Vehicles Be Safe and Trustworthy? Effects of Appearance and Autonomy of Unmanned Driving Systems". In: International Journal of Human–Computer Interaction 31.10 (2015), pages 682–691

Alastair Faulkner, Abbeymeade Limited

Alastair has over 40 years' experience in the application of systems, software and safety engineering. The last 20 years have been in safety consultancy. An early interest in data dependent systems led to an Engineering Doctorate at Warwick (*Data integrity: an often ignored aspect of safety systems*). Alastair is a highly competent practitioner, with comprehensive experience in senior development and management roles in systems, applications, and infrastructure environments. Alastair is a joint author of an Elsevier publication '*Data-centric Safety – Challenges, Opportunities and Incident Investigation*' will be available from March 2020 – ISBN 978-0-12-820790-1) with Mark Nicholson.

Email: alastair.faulkner@abbeymeade.co.uk