REDUCING OUR IGNORANCE: FINDING ANSWERS TO CERTAIN EPISTEMIC QUESTIONS FOR SOFTWARE SYSTEMS

C. M. Holloway*, C. W. Johnson †

**NASA Langley Research Center, Hampton, Virgínia, USA, C.Michael.Holloway@nasa.gov*
†Dept. of Computing Science, University of Glasgow, Scotland, UK, Christopher.Johnson@glasgow.ac.uk

Keywords: safety, accidents, epistemology, requirements, confidence.

Abstract

In previous papers, we asserted that software system safety is primarily concerned with epistemic questions, that is, questions concerning knowledge and the degree of confidence that can be placed in that knowledge. We also enumerated a set of 21 foundational epistemic questions, discussed some of the difficulties that exist in answering these questions adequately today, and speculated briefly on possible research that may provide improved confidence in the sufficiency of answers in the future. This paper focuses on three of the foundational questions. For each of these questions, current answers are discussed and potential research is proposed to help increase the justifiable level of confidence.

1 Introduction

Begin at the beginning and go on till you come to the end; then stop.
(Lewis Carroll, Alice in Wonderland)

Philosophy and engineering seem to be very different, unrelated disciplines. Philosophy asks grand, abstract, big picture questions. Why is there something rather than nothing? What is real? What is truth, and how do we know when we have found it? Engineering asks ordinary, practical, detailed questions. How much weight can the bridge hold? Does this particular wing design provide enough lift? How can we build a stronger, more effective, cheaper mousetrap? Some branches of philosophy are irrelevant to engineering, but at least one—epistemology—is quite relevant.

Epistemology is one of the major branches of study in philosophy [8]. It is concerned with seeking answers to questions such as, 'What do we know,' and 'How do we know we know what we know?' [26]. Abstract questions such as these may have little direct relevance to engineering, but concrete versions of such questions are not only relevant, but essential.

Consider, for example: What do we know about the safety of an automated control system? Upon what assumptions are we resting our knowledge? How do we know that what we think we know about safety is accurate? These are critical engineering questions; for system safety professionals, they are probably the most important questions. They are also questions about epistemology (that is, epistemic questions).

In two previous papers [16, 17], we identified and refined a collection of fundamental epistemic questions about software system safety, and discussed possible research activities that may enable improved confidence in the accuracy of answers to the questions. In this paper, we focus on three specific questions:

- *What does 'at least as safe as' mean?*
- *What is the appropriate level of confidence to be attached to the satisfaction of standards?*
- *What information is available to accident investigators?*

After a synopsis of our previous work, we devote one section to each of the three questions. In the final section of the paper, we address another question ('So what?') by suggesting some ways in which the ideas presented in the paper may benefit its readers.

2 Previous Work

The past is never dead. It's not even past.
(William Faulkner, Requiem for a Nun, Act I, Scene iii)

Readers who are familiar with the two previous papers may skip this section. Readers who are not familiar with those papers should find enough information here to make the rest of the paper intelligible on its own.

2.1 Motivation

For any system upon which lives depend, the system should not only be safe, but the designers, operators, and regulators of the system should also know that it is safe. For software-intensive systems, universal agreement on what is necessary to justify knowledge of safety does not exist. Theorists and practitioners have long quarrelled with each other and among themselves over the issue.

The existence of these quarrels, the associated wide range of existing opinions among well-known experts, and the emotional fervour with which these opinions are held and

expressed [18, 33] motivated our initial work. We posited that one of the chief reasons for the lack of consensus may be that the community is trying to answer broad questions without first refining those questions into simpler, more foundational questions. So we set out to discover what those foundational questions might be.

2.2 Definitions

For these foundational questions to be fully understood, a common understanding of certain words and phrases is needed.

Epistemology was already described in the introduction. Its primary definition is 'the theory or science of the method or grounds of knowledge'. A common related adjective is *epistemic*, which means 'of or relating to knowledge or degree of acceptance' [29].

Common verbs used in relation to epistemic questions include *believe*, *think*, and *know*. These verbs have multiple shades of meaning, and are often used somewhat differently by different people. One person may use the three verbs almost interchangeably. Another person may use the three words to express graduated levels of confidence. For such a person, *believe* may correspond to 'more likely than not', *think* to 'very likely', and *know* to 'beyond a reasonable doubt' (or perhaps to even a stronger standard)[1].

Safety may be defined absolutely as 'freedom from accidents or losses' [23], with the adjective *safe* thus similarly meaning 'free from accidents or losses.' Such definitions are ideals; no system can be truly said to be absolutely and forever free from accidents or losses until it has been decommissioned. So, in practice the words are implicitly (if not always explicitly) modified by a notion of acceptability; where the definition of *acceptable* varies over time, among different domains and systems, among different regions and cultures, and even among different individuals [6], [22], [37], [40], [41].

Based on the above definitions, the sentence that began section 2.1 means 'For any system upon which lives depend, the system should not only be acceptably free from accidents and losses, but the designers, operators, and regulators of the system should also have the required level of confidence that the system is acceptably free from accidents and losses.'

2.3 Results

We developed a collection of 21 foundational epistemic questions, divided into two primary categories: questions about existing systems, and questions about future systems. The questions about existing systems were further divided into two additional categories: questions about that are independent of whether an accident or loss has occurred to the system and questions that are specific to gaining knowledge after an accident or loss has occurred. The questions about future systems were divided into three additional categories: questions about systems that are intended to replace existing operational systems, questions about systems that are truly new, and questions that are common to both replacements and truly new systems.

The 21 questions are shown below in the order in which they were numbered and described in our second paper [17][2]:

Existing Systems: Accident-Independent
1 How is operational safety assessed?
2 How does operational safety compare with expected safety?
3 How should difference in safety assessments be reconciled?
4 How does the operational environment affect safety?
5 What maintenance is required for safety?
6 How do changes affect safety?

Existing Systems: Accident-Related
7 What information is available to accident investigators?
8 How do investigators know all relevant factors have been found?
9 How can lessons taught by an accident improve safety?

Future Systems: Replacements
10 What does 'at least as safe as' mean?
11 What are the safety implications during transition?

Future Systems: Truly New
12 How is the desired level of safety to be determined?
13 What can be learned from existing systems?
14 How will novel technologies affect safety?

Future Systems: Both Replacements and Truly New
15 What level of confidence in safety is required?
16 How is knowledge obtained about the intended operational environment?
17 How is the sufficiency of safety requirements assured?
18 How is the sufficiency of implementation assured?
19 How are assumptions and implications understood?
20 What is the level of confidence provided by assessment methods and tools?
21 What is the appropriate level of confidence to be attached to the satisfaction of standards?

Questions 10, 21, and 7 are the focus of the reminder of the current paper. Each is discussed in a separate section below.

[1] Some philosophers enjoy debating whether applying the verb 'know' (or the noun 'knowledge') to something that is false is ever appropriate [5], [9], [13]. As fun as such a debate may be, it is irrelevant in discussions about system safety: words such as 'know' and 'knowledge' are used, and sometimes that which is said to 'known' turns out to be false.

[2] Some readers may be asking, 'Is there any significance to the numbering scheme, or the order of the questions?' The answer is, 'No.' Although developing a priority scheme for the questions, and numbering them according to that scheme is possible, perhaps even desirable, we have not done it.

3 As Safe As

Out of this nettle, danger, we pluck this flower, safety.
(William Shakespeare, Henry IV Part 1, Act II, Scene iii)

A common requirement imposed on any new system that will be used to replace an existing system is that it *be at least as safe as* the system it is replacing. This requirement raises the obvious question: *What does 'at least as safe as' mean?* Or, to rephrase the question into a more explicitly epistemic form: *How can we know that a new system will be 'at least as safe as' the system it will replace?*

3.1 Today

Answers to the question today vary depending on specifics about the application domain, the systems themselves, and the operating environment. Equally varying is the level of confidence that can be justifiably placed in the answers.

On one side of the similarity spectrum are systems with characteristics such as the following:

- the operating environment for the replacement system will be the same as for the old system;
- the functional requirements are unchanged;
- the new system will use identical components to the old system, within an identical architecture;
- the behavior of system components individually and collectively is well-understood;
- analyses and tests conducted on the old system before it was deployed correspond well to actual operation, and the same analyses and tests will be conducted on the new system.

As long as state-of-the-practice design, assessment, and testing, and system safety methods are followed, a very high level of confidence is justified that a new system with these characteristics will meet or exceed the operational safety of the system it is replacing.

In addition to not having any of the positive characteristics listed above, replacement systems on the far other side of the similarity spectrum have characteristics such as these:

- the new system implements functions in software that either did not exist in the old system or that were implemented by other means;
- new technologies will be used in some aspect of the replacement system;
- the interface between the operators and the system is different;
- the projected operational costs for the new system must be substantially less than those of the old system.

If a replacement system possesses all of these characteristics, then it is fundamentally and substantially different from the existing system. In such a situation, 'at least as safe as' would not be so much a specific requirement as it would be one of the considerations used in determining the desired level of safety, and answering questions 12-14 would be more important than answering question 10.

Most replacements systems will have characteristics that fall between the extremes. A common scenario is one in which the replacement system is intended to share many of the same components and general architecture with the old system and perform all of the functions that the old system could perform (perhaps with some of these functions newly implemented in software), while also performing some entirely new functions. The introduction of anti-lock braking systems (ABS) to automobiles was an instantiation of this scenario, for example. ABS perform all of the functions of traditional braking systems, while adding the function of preventing lock-up of the brakes (and the attendant loss of steering control) when rapid deceleration is required.

Conversations with industry practitioners suggest that current best practices for satisfying the 'at least as safe as' requirement are based on comparative hazard analyses. That is, a planned replacement system will be considered to satisfy the requirement if it can be shown to eliminate or control all the hazards eliminated or controlled by the existing system, without introducing any new, inadequately handled hazards. The difficulty of making such a demonstration (and the confidence that can be placed in the correctness of the demonstration) is affected by the same factors that affect the difficulty of hazard analysis in general. Further complexity arises when the introduction of a new system creates new modes of operation or encourages methods of working that cannot directly be compared with previous operations. For instance, some have argued that the introduction of ABS systems encourages drivers to brake later and harder knowing that they are protected by the additional safety margins provided by these devices [41].

3.2 Tomorrow

Thus, the research that will permit higher levels of justified confidence in answering the 'as safe as' question is similar to the research that will improve confidence in the completeness and efficacy of hazard analysis generally. Potential research areas include the following:

- identifying novel hazards arising from new technologies, applications, domains, or environments;
- understanding hazards introduced by choices made for human-machine interfaces;
- predicting the consequences and likelihoods of worst-case events;
- recognizing when assumptions made during system design have been violated in system operation to the extent that new hazards may exist which the system was not designed to eliminate or control;
- transferring lessons taught concerning hazards in one domain or industry to other domains or industries;
- exploring architecturally-based hazard mitigation strategies for software-intensive systems;

- increasing automated support for all of the above areas.

One promising approach to conducting research in these areas is to exploit the natural relationship between after-the-fact accident causality relationships and before-the-fact hazard identification. An example of this approach is Leveson's System-Theoretic Process Analysis [25].

4 Standards & Confidence

We are generally the better persuaded by the reasons we discover ourselves than by those given to us by others.
(Blaise Pascal)

Standards are a controversial topic, particularly in relation to software-intensive systems, and even more particularly in relation to the safety of software-intensive systems. Question 21 from our list engages the controversy directly: *What is the appropriate level of confidence to be attached to the satisfaction of standards?*

4.1 Today

Much current debate revolves around how this question should be answered. Significant differences of opinion exist concerning the relative importance of controls on the process used to develop software, satisfaction of pre-determined standardized objectives for each software system, and the development of system-specific safety arguments.

Existing standards embody the full range of these differences, from the fully process-centric [38] to the completely argument-centric [27], and everything in between [1, 2, 32, 34, 35, 36]. Amplifying the differences of opinion are published criticisms of various approaches contained not only in academic and professional papers [7, 10, 15, 40], but also in reports of government-appointed reviews [14, 28] and prestigious technical bodies [19]. Decreasing the likelihood of resolving the differences are legal and business reasons for companies to keep secret the details of exactly how they satisfy applicable standards.

The situation is such that for any specific instantiation of the question ('What is the appropriate level of confidence to be attached to the satisfaction of specific standard X?'), well-known, respected people can likely be found to answer, 'None at all.' Equally well-known and respected people can also likely be found to answer, 'Beyond a reasonable doubt.'

4.2 Tomorrow

To improve the current chaotic state of affairs in this area, a combination of both research and social / cultural changes seems necessary. This combination includes items such as the following:

- Conducting retrospective studies of the effectiveness of existing methods, processes, tools, and standards as actually used;
- Developing a better understanding of how various standards are applied in practice, rather than basing criticisms on the bare text alone;
- Developing standards only after credible evidence exists to support the standard;
- Conducting case studies and realistic experiments on new methods, processes, and tools before they widely used;
- Removing legal and business barriers to open sharing of any type of information that relates to improving safety;
- Exploring ways to make safety arguments for public systems publically available in an understandable form.

Properly developed, applied, and maintained standards have contributed significantly to quality and safety in many disciplines. Activities and research such as proposed above should help ensure that this will be, in reality and perception, the case for software systems-related standards, too.

5 Aiding Investigators

I'm not saying there won't be an Accident now, mind you. They're funny things, Accidents. You never have them till you're having them.
(A. A. Milne, The Complete Tales of Winnie the Pooh)

History suggests that for nearly any complex system, accidents of some sort are inevitable, no matter how carefully the system is design, deployed, and operated. When an accident happens, the likelihood that investigators will be able to determine what happened is strongly related to the quantity and quality of the information available to them. Hence the question: *What information is available to accident investigators?* Or, to rephrase the question into a more explicitly epistemic form: *How do accident investigators know they have uncovered all the relevant information?*

5.1 Today

For traditional systems and components a good understanding tends to exist of the information needed to reach valid conclusions in an accident investigation. Flight data and voice recorders, for example, are required for commercial aircraft, because the information they contain is often critical to an accident investigation. An illustration of the criticality of the information contained in such devices can be seen in the June 2009 crash of Air France in the Atlantic Ocean off the coast of South America. Before the flight data and voice recorders were found in May 2011, the causes of the accident were almost certainly not going to discovered by the investigating agency, France's Bureau d'Enquêtes et d'Analyses pour la sécurité de l'aviation civile (BEA). Now, however, the BEA is significantly more confident that they will be able to determine what happened [11].

The situation for software-intensive systems is different from the situation for more traditional systems. No consensus has yet been reached about what constitutes the necessary or sufficient information to be recorded and preserved for accident investigators, even within the commercial air transportation domain. Some specific systems or subsystems keep track of sufficient information to enable investigators to

recreate the state of the software and computers at the time of an accident[3]. Other systems or subsystems do not. So the current answer to the first form of the question is, 'It depends,' and the answer to the second form is, 'They might not.'

5.2 Tomorrow

To move these answers from the doubtful to the affirmative, research in the following areas seems needed:

- Exploring the efficacy and practicality of currently proposed ideas for providing more information to investigators, such as cockpit video recorders, and real-time streaming of data;
- Creating techniques and tools for assisting software system implementers in determining what information about these systems should be made available in the event of an accident;
- Determining how to ensure that this information is preserved, and presented to investigators in a usable format;
- Developing and evaluating accident causation models [4] that account adequately for the rapidly increasing complexity of software-intensive systems;
- Evaluating the extent to which lessons taught by previous accidents are being learned and incorporated into new systems.

Without research in these areas, investigating accidents involving software-intensive systems is likely to continue to be difficult, with confidence in the results often lower than desired.

6 A Final Question: So What?

An idea not coupled with action will never get any bigger than the brain cell it occupied
(Arnold H. Glasgow)

This paper expanded on three of twenty-one epistemic questions about software systems. Researchers who read the paper may want to consider whether their expertise and interests match any of the proposed research areas, and if so, how they might begin to make contributions in the area(s). Industry practitioners may want to consider how their companies answer the three questions, and whether there is anything that they can do to improve those answers. Regulators may want to consider whether one or more of the ideas presented here might help them in their daily activities. Readers of all types may want to consider reading the references cited in the paper. Everyone is welcome to correspond with us about this work.

[3] See [3] for an example of an incident in which sufficient information was available to investigators. However, it is not clear from published accounts, or from private conversations with individuals familiar with unpublished details about the occurrence, whether adequate information would have been available if the identical events had resulted in a hull loss, instead of simply a bit of internal damage, and a very unpleasant ride for the passengers.

References

[1] Australian Government. DEF(AUST)5679 / Issue 2. Safety Engineering for Defence Systems, (2008).

[2] Australian Government. DEF(AUST)10679 / Issue 1, Guidance Material for DEF(AUST)5679 / Issue 2, (2008).

[3] Australian Transport Safety Bureau. *In-flight upset event 240 km north-west of Perth, WA Boeing Company 777-200, 9M-MRG, 1 August 2005,* Aviation Occurrence Report 200503722 Final, (2007).

[4] Australian Transport Safety Bureau. *Analysis, Causality and Proof in Safety Investigation,* Aviation Research and Analysis Report: AR-2007-053, (2008).

[5] Bahnsen. G. "A Conditional Resolution of the Apparent Paradox of Self-Deception", Ph.D. dissertation., University of Southern California, (1978).

[6] Barley, S. *The Search for Air Safety: An International Documentary Report on the Investigation of Commercial Aviation Accidents*. William Morrow & Company, Inc., (1970).

[7] Caseley, P.R. and T.A.D. White. "The MOD Procurement Guidance on Software Safety Assurance – Assessing and Understanding Software Evidence", *Proceedings of the IET 4th International Conference on System Safety*, (2009).

[8] Clark, G. H. *Thales to Dewey*. Trinity Foundation (1989).

[9] Damar, T. E. *Attacking Faulty Reasoning: A Practical Guide to Fallacy-Free Arguments.* 5th edition. Thomson-Wadsworth, (2005).

[10] Fenton, N. E., Neil, M. "A Strategy for Improving Safety Related Software Engineering Standards", IEEE Transactions on Software Engineering, **24**(11), pp. 1002-1013, (1998).

[11] Flottau, J., Wall, R. "BEA Releases Initial Sequence of Events in AF447 Crash", *Aviation Week & Space Technology*, p. 36, June 6, (2011).

[12] Greenwell, W. S. Pandora: An Approach to Analyzing Safety-Related Digital-System Failures, Ph.D. thesis, School of Engineering and Applied Sciences, University of Virginia, (2007).

[13] S. Haack. *Defending Science — within reason*. Prometheus Books, (2007).

[14] Haddon-Cave, C. *The Nimrod Review: An Independent review into the broader issues surrounding the loss of*

the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006. London: The Stationery Office. (2009).

[15] Hawkins, R. D., Kelly, T. P. "A Systematic Approach for Developing Software Safety Arguments", *Proceedings of the 27th International System Safety Conference*, Huntsville, Alabama, (2009).

[16] Holloway, C. M and Johnson, C. W. "Towards a Comprehensive Consideration of Epistemic Questions in Software System Safety", *Proceedings of the IET 4th International Conference on System Safety*, London, U.K., (2009).

[17] Holloway, C. M. and Johnson, C. W. "Epistemic Questions & Answers for Software System Safety", *Proceedings of the 28th International System Safety Conference*, Minneapolis, Minnesota, (2010).

[18] Holloway, C. M; Johnson, C. W.; Collins, K. R. "A Safety Conundrum Illustrated: Logic, Mathematics, and Science are not Enough", *Proceedings of the IET 5th International Conference on System Safety*, Manchester, U.K., (2010).

[19] Jackson, D., Thomas, M., Millett, L. I. (eds). *Software for Dependable Systems: Sufficient Evidence?* National Research Council, Committee on Certifiably Dependable Software Systems, (2007).

[20] Jet Propulsion Laboratory, JPL Special Review Board. "Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions", JPL D-18709, (2000).

[21] Johnson, C. W., Holloway, C. M. "The Dangers of Failure Masking in Fault Tolerant Software: Aspects of a Recent In-Flight Upset Event", *2nd IET International Conference on System Safety*, London, (2007).

[22] Johnson, C.W. *Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting*. University of Glasgow Press, Glasgow, Scotland, United Kingdom, (2003).

[22] Leveson, N. G. "High Pressure Steam Engines and Computer Software", *IEEE Computer*, **27** (10), pp. 65-73, (1994).

[23] Leveson, N. G. *Safeware: System Safety and Computers*. Addison-Wesley, (1995).

[24] Leveson, N. G. "Applying Systems Thinking to Analyze and Learn from Events", *NeTWorK 2008: Event Analysis and Learning from Events*, Berlin, (2008).

[25] Leveson, N. G. *Engineering a Safer World: Systems Thinking Applied to Safety*. <http://sunnyday.mit.edu/safer-world/> To be published by MIT Press, (2011).

[26] McCarthy, N. "Philosophy and engineering", *Inter-disciplinary Science Reviews*, **33**, No. 3, (2008).

[27] Ministry of Defence. "Safety Management Requirements for Defence Systems", Defence Standard 00-56, Parts 1 and 2, Issue 4, (2007).

[28] National Commission on the BP Deepwater Horizon Oil Spill and Offshore Drilling. *Deep Water: The Gulf Oil Disaster and the Future of Offshore Drilling*, Report to the President, (2011).

[29] *The Oxford English Dictionary Online*, Oxford University Press, <http://www.oed.com/>, (2011).

[30] Petroski, H. *Design Paradigms: Case Histories of Error and Judgement in Engineering*. Cambridge University Press, (1994).

[31] Petroski, H. *To Engineer is Human: The Role of Failure in Successful Design*. Vintage Books, (1992).

[32] RTCA/EUROCAE. "Software Considerations in Airborne Systems and Equipment Certification", DO-178B/ED-12B, (1992).

[33] Safety Critical Mailing List Archive. www.cs.york.ac.uk/hise/safety-critical-archive/ (2011).

[34] Society of Automotive Engineers. *Certification Considerations for Highly-Integrated or Complex Aircraft Systems*, SAE ARP 4754, (1996).

[35] Society of Automotive Engineers. *Guidelines for Development of Civil Aircraft and Systems*, SAE ARP 4754a, (2010).

[36] Society of Automotive Engineers. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, SAE ARP 4761, (1996).

[37] Snook, S. A. (2000). *Friendly Fire: The Accidental Shootdown of U.S. Black Hawks over Northern Iraq*. Princeton University Press.

[38] Software Engineering Institute. *CMMI[SM] for Software Engineering (CMM-SW, V1.1)*. CMU/SEI-2002-TR-029. (2002).

[39] Vaughan, D. *The Challenger Launch Decision*. The University of Chicago Press, (1996).

[40] Weaver, R. A. The Safety of Software - Constructing and Assuring Arguments. PhD thesis, Department of Computer Science, The University of York (2003).

[41] Wilde, G. J. S. *Target Risk 2: A new psychology of safety and health*. (2001).