Epistemic Questions & Answers for Software System Safety

C. M. Holloway; NASA Langley Research Center; Hampton, Virginia, USA

C W. Johnson; University of Glasgow; Glasgow, Scotland, UK

Keywords: system safety, safety case, argument, evidence, epistemology

Abstract

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. For systems with which human experience is long, such as roads, bridges, and mechanical devices, knowledge about what is required to make the systems safe is deep and detailed. High confidence can be placed in the validity of that knowledge. For other systems, however, with which human experience is comparatively short, such as those that rely in part or in whole on software, knowledge about what is required to ensure safety tends to be shallow and general. The confidence that can be placed in the validity of that knowledge is consequently low. In a previous paper, we enumerated a collection of foundational epistemic questions concerning software system safety. In this paper, we review and refine the questions, discuss some difficulties that attend to answering the questions today, and speculate on possible research to improve the situation.

Introduction

On first thought, one might be hard pressed to think of two disciplines less alike than philosophy and engineering. Philosophy is concerned with 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 is concerned with 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? This first thought is probably correct for some aspects of philosophy; but for at least one—epistemology—it is quite wrong.

Epistemology is one of the major branches of study in philosophy (Clark, 1989); it is concerned with searching for answers to questions such as, ‘What do we know,’ and ‘How do we know we know what we know?’ (McCarthy, 2008). Although abstract questions such as these may have little direct relevance to engineering, concrete versions of such questions are not only relevant, but essential.

Consider, for example: What do we know about the safety of this automated control system? How do we know that what we think we know about its 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 a previous paper (Holloway & Johnson, 2009), we identified an initial collection of fundamental epistemic questions about software system safety. In this paper, we explain necessary context and definitions, present a refinement of the previously developed questions, discuss some difficulties that exist in answering the questions today, and speculate on possible research that may enable improved confidence in the accuracy of answers to the questions.

Motivation & Definitions

Consider two questions a system safety engineer may ask about a particular system: 'Is the system safe?' 'Do I know the system is safe?' In an ideal world for any specific system, the answer to the first question is the same as the answer to the second. That is, if we *know* the system is safe, then it *is* safe; and if we do not know the system is safe, then it is not safe. But the real world is not an ideal world. In the real world the answers to the two questions may differ. We may know a system is safe when it is not, and we may ‘know’ a system is not safe when it is.

The potential difference in the answers to the two questions is especially apparent today in software-intensive systems. While many software safety experts lament the lack of adequate means for assessing the safety of software systems, denounce existing software standards as based on weak or non-existent foundations, and warn against increasing reliance on automated systems, the actual safety record of software-based systems has been exceptionally good to date. So good in fact, that a strong case can be made, at least for commercial aviation, that no technology yet introduced has had a more positive effect on safety than has software. On the other hand, despite the excellent safety record to date, the arguments about future dangers seem quite persuasive, particularly as systems become increasingly complex, and more and more authority is given to automated systems to perform safety-critical functions.

Trying to understand why this discrepancy exists between current practice and theory provided the initial motivation for exploring epistemic questions. The definitions that follow provide the context for understanding the results of this exploration so far.

Definitions Concerning Knowledge: *Epistemic* is an adjective meaning 'of or relating to knowledge or degree of acceptance' (Oxford University Press, 1989). *Epistemology* is a noun defined as 'the theory or science of the method or grounds of knowledge' (Oxford University Press, 1989). Epistemology is one of the major branches of study in philosophy (Clark, 1989); it is concerned with searching for answers to questions such as, 'What is knowledge,' and 'How is knowledge acquired?'

The verbs *believe*, *think*, and *know*, which are used in relation to knowledge, all have multiple shades of meaning, and tend to be used somewhat differently by different people. One person may use the three verbs almost interchangeably. For such a person, these three questions are essentially identical: Do I believe the system is safe? Do I think the system is safe? Do I know the system is safe?

Another person may use the three words to express graduated levels of confidence. For such a person, the three questions are quite different; answering them affirmatively requires different levels of personal certainty in the safety of the system. For example, *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). For the purposes of this paper, we adopt this level-of-confidence based approach[1].

Regardless of how a particular individual uses the three verbs, he or she may be wrong. For example, Jill may believe, think, or know that the 28th International System Safety Conference is being held in Yorktown, Virginia, in October 2010. The strength of Jill's level of confidence does not change the fact that she is simply wrong (Bahnsen, 1978; Damar, 2005; Grudy 2005).

Definitions Concerning Safety: The noun *safety* can be defined absolutely as 'freedom from accidents or losses' (Leveson, 1995), with the adjective *safe* thus similarly meaning 'free from accidents or losses.' Such definitions are recognized to be ideals, which are not fully achievable in practice. No system can be truly said to be absolutely and forever free from accidents or losses. So, in practice the words tend to be used relativistically. Commercial air travel is said to be safe, for example. This attribution of safety does not mean that *no* accidents or losses ever occur in commercial air travel, but that accidents and losses occur with sufficient rarity as to be considered acceptable.

Understanding the practical definition of safety thus requires understanding the meaning of *acceptable*. What degree of freedom from accidents and losses is acceptable? Answers to that question have varied over time, among different domains, among different regions of the world, and even among different individuals (Leveson, 1994; Petroski, 1992; Wilde, 2001).

In the context of system safety, these variations may be subsumed by an operational definition of acceptability for each system. For commercial air travel, the acceptability of its current level of freedom from accidents and losses is seen in the combination of the facts that users continue to fly, engineers and companies continue to produce aircraft and other components necessary for air travel, regulatory bodies continue to produce regulations for air travel, and governments continue to allow air travel within their boundaries. No single one of these facts taken alone necessarily implies acceptable safety, but taken together they do.

[1] Although we *know* that any philosopher reading this paper will consider this section simplistic and incomplete, we *believe* that it is sufficiently detailed and complete for the intended audience. The strength of this belief is higher for the current paper than it was for the previous one, based on comments from philosophers and non-philosophers on a similar section in the first paper.

Based on the above definitions, the first question that opened this section ('Is the system safe?') may be understood to be equivalent to 'Is the system acceptably free from accidents and losses?' Adopting the confidence-level-based definitions for believe, think, and know, and assuming that for safety-critical systems, the highest level of confidence is required, the second question ('Do we know the system is safe?') means 'Do we have confidence at least beyond a reasonable doubt that the system is acceptably free from accidents and losses?'

The remainder of the paper concerns this latter question. For simplicity of expression, we often revert to the shorter form, relying on the reader to mentally translate to the longer form.

Foundational Epistemic Questions

For any system upon which lives depend, the system should not only *be* safe, but the designers, operators, users, and regulators of the system should also *know* that it is safe. For software-intensive systems, a consensus does not exist on what is necessary to constitute knowledge. Theorists and practitioners have long quarreled with each other and among themselves over the issue. The wide range of existing opinions, and the emotional fervor with which these opinions are held (Safety Critical Mailing List Archive, 2010), suggests that reaching a consensus is not soon likely.

Perhaps one of the reasons for the lack of consensus is that the community is trying to answer the broad questions, without first refining those questions into more foundational questions. Such a situation is analogous to a jury in a criminal trial trying to answer the ultimate question, 'Is the defendant guilty,' without first answering questions whose answers provide evidence upon which to base the ultimate answer. Questions such as, 'Was the defendant present at the scene of the crime', 'Did the defendant have the means to commit the crime', and 'Could someone be trying to frame the defendant?'

In creating our initial list of foundational questions for software system safety, we considered two main categories: existing systems, and future (yet-to-be-built) systems. We continue to believe that these are appropriate top-level categories; each is discussed below, with additional sub-categories identified, and specific questions listed. The format used for listing the questions is to number them sequentially, give a short form of the question, discuss what the question means (with perhaps additional related questions), and mention some of the difficulties involved in answering it. Where appropriate, we also discuss potential areas for fruitful research.

Questions About Existing Systems: In the original paper, we distinguished between two categories of existing systems: systems that have been operating for sufficiently long that they are 'known' to be safe, and systems that have not been operating that long. Because all but a couple of epistemic questions are similar for both of these categories, we no longer consider that particular distinction to be important. Instead we divide the questions into those that are independent of whether an accident or loss has occurred and those that are specific to gaining knowledge after an accident or loss has occurred.

Accident-independent questions: There are at least six[2] foundational epistemic questions that can be asked about all existing systems.

(1) *How is operational safety assessed?* This question is intended to determine, for each specific system, the method or methods used to decide that the system is acceptably free from accidents and incidents as it is used. Related questions include the following. What is necessary for a system to be considered to have its safety effectively demonstrated? What information must be collected and analyzed to provide adequate confidence in the continuing accuracy of a safety assessment? For how long must this information be collected and analyzed?

Answering these questions is complicated by the difficulty of gathering accurate operational data in many industries. Without access to such data, knowing how close an outwardly safe system may be to having an accident is difficult. A system may appear to be functioning in a fully safe and successful manner for a long period of time, but it may in fact be only a few small steps away from a major accident (Johnson, Herd, Wolff, 2010). Improving operational data collection will involve not only research into non-obtrusive (to both humans and machines) means of collection, but also resolution of various ethical and legal issues that arise.

[2] The original paper also listed six; however, we have combined two very similar questions from the original list, and moved one question to this section from another section.

(2) *How does operational safety compare with expected safety?* This question is intended to help safety engineers and others determine whether the pre-deployment predictions of the expected safety of a particular system adequately predict the actual safety of that system once it is deployed. Consistently asking and answering the question might result in a better understanding of the efficacy of system safety procedures and tools. It might also prevent some accidents and losses from happening, because history shows that accidents sometimes occur after a period in which the operational performance of a particular system has deviated from its expected performance in ways that were not realized at the time (Haddon-Cave, 2009; Snook, 2000; Vaughan, 1996). Answering the question is not an easy task, however, for the reasons just discussed above.

(3) *How should difference in safety assessments be reconciled?* For example, consider a software-intensive medical device, which is considered safe by the appropriate regulatory authority, but which has occasionally failed in such a way as to lead to successful lawsuits against its manufacturer. What should be done in this case? What evidence is needed to permit an informed decision to be made by the regulatory authority? From one perspective, the argument can be made that the existence of a flaw increases the likelihood of other undiscovered safety-reducing problems, and thus the device should be removed from the market. From another perspective, the argument can also be made that the discovery and rectification of the flaw increase confidence that the device will never fail in that way again.

(4) *How does the operational environment affect safety?* This question is particularly relevant for systems that come to be used in environments different from those anticipated when the systems were originally designed. Asking and trying to answer the question will help improve assessments of the potential safety consequences of changes in the operational environment. Answering the question can be difficult because it can be very hard to identify all of the myriad specific changes in an operational environment that should legitimately affect confidence in the safety of the system. Research into better methods and tools for identifying safety-relevant environmental factors seems potentially fruitful.

(5) *What maintenance is required for safety?* Any system involving components that can degrade over time requires maintenance to ensure that such components are replaced or repaired before their degradation negatively affects system safety. The extent, frequency, and criticality of this maintenance needs to be well understood, as does the means for ensuring that required maintenance is performed properly and on time. Because maintenance is also known to be a factor in causing accidents, maintaining a system may both bolster and undermine confidence in safety.

(6) *How do changes affect safety?* Once in operation, few systems remain unchanged. Whether in response to accidents (as discussed in the next section), or to modify or extend functionality, changes are common, particularly for software-intensive systems where making changes appear to be deceptively simple. When changes are made to an operational system, those changes should not adversely affect the safety of the system. Ensuring this is the case requires asking the appropriate question and determining the level of confidence required in the answer to it. Because software is increasingly being used to tailor or configure complex systems in a variety of industries, operating profiles for a system may be subject to almost daily changes. In situations such as these, determining the effect that a change should have in confidence of safety can be very difficult; the magnitude or extent of a change may or may not have proportionate impact on knowledge of the safety of a such a complex system. Research in change-impact analysis is needed to help answer this epistemic question with more confidence than is possible now.

Accident-related questions: As mentioned earlier, and as any system safety professional understands, no system is perfectly safe. So for almost any real system, an accident[3] will eventually occur. When this happens, there are at least three[4] epistemic questions that should be asked and answered.

(7) *What information is available to investigators?* The likelihood that an accident or incident investigation will be able to determine what happened is strongly related to the quantity and quality of the information available to them. Over time engineers and investigators have come to have a good understanding of the sorts of information needed for traditional systems and components. The situation is different for software-intensive systems, where a consensus

[3] We use 'accident' as a shorthand. Consider it to include accidents, incidents, and other undesirable outcomes for which investigations are conducted.

[4] The original paper listed six. Question (8) discussed here was divided into four separate, but closely related questions.

about the necessary information has yet to be reached. An expanded version of the fundamental question is as follows: What information about the system and its state at the time of the accident must be available to investigators to enable them to gain sufficient knowledge to be able to conduct a thorough investigation? A related question is What do investigators do if adequate information is not available? See (Jet Propulsion Laboratory, 2000) for example of a situation in which investigators had to make do with inadequate information.

Answering this question can be complicated by the extensive fault-tolerance mechanisms used in many advanced systems. Software or hardware errors may be caught and effectively handled by such mechanisms, which can help raise legitimate confidence in safety. However, these mechanisms can also mask the source of some types of failures from system operators, which may make recovering from those failures more difficult than it otherwise would be (Johnson & Holloway, 2007). Regardless of whether system operators have access to all the failure information, accident investigators must have access to the information so that they are able to identify the true factors leading to certain accidents (Australian Transport Safety Bureau, 2007).

(8) *How do investigators know all relevant factors have been found?* The subject of causality is one of strong interest, both theoretically (Collins, Hall, and Paul, 2004) and practically (Australian Transport Safety Bureau, 2008). For the purposes of improving safety of the particular system involved in the accident and of related or potential future systems, investigators need to identify all the factors that contributed to making the accident happen, and they need to have a high level of confidence that they have done so. As systems become more complex, and more reliance is placed on software systems with which little previous experience exists, this task becomes more difficult, and confidence that it has been completed subsequently lower.

(9) *How can lessons taught by an accident improve safety?* Identifying causes alone is unlikely to improve safety. This information must be used to develop improvements to prevent these causes from leading to another accident, not only in the particular system investigated, but in similar systems, either currently operating or ones to be developed in the future. These improvements must be implemented, and the knowledge that led to their creation must be made available in an understandable form for as long as it is relevant. Also, designers and engineers of future systems not only need to have access to the lessons taught by previous accidents, they must also have the motivation to seek out and use this information. Too often designers and engineers seem not to read full accident reports, but instead rely on word of mouth, which may convey partial and biased views about the causes of previous accidents (Holloway & Johnson, 2006).

These three questions, and possible ways to answer them, have been considered in various ways—see for example (Collins, Hall, and Paul, 2004; Greenwell, 2007; Holloway, 1999; Johnson, 1997; Johnson, 2003; Leveson, 1994; Leveson, 1995; Petroski, 1992; Petroski 1994)—but we are unaware of any systematic, detailed research efforts aimed towards developing methods for providing cogent, comprehensive answers to all of them. The creation and successful execution of such a research program could make a very important, potentially life-saving contribution to system safety world-wide.

<u>Questions About Future Systems</u>: As difficult to answer as questions about existing systems may be, the foundational epistemic questions about systems that have not yet been fielded may be even more difficult to answer. These future systems can be divided into two main categories: systems that are intended to replace existing operational systems; and systems that are truly new. The two categories share some epistemic questions, and have some unique ones also. We discuss the questions unique to replacement systems first, then those unique to truly new systems, and conclude with those common to both categories of systems.

Replacement system questions: A common type of new system is one that is intended to replace a system that is already in place. Many different reasons may exist for creating such a replacement system, ranging from introducing software systems where there were none previously (as has been done, for example, in aircraft with the introduction of fly-by-wire flight controls), to upgrading the capabilities of an existing software system (for example, modernization of various air traffic management systems). Regardless of the motivation for the replacement, at least two[5] epistemic questions should be asked.

[5] Reduced from four in the original work.

(10) *What does 'at least as safe as' mean?* The most common safety requirement imposed on a replacement system is that it be at least as safe as the system it is replacing. Such a requirement seems quite reasonable in theory, but in practice determining what it means may be difficult. Part of that determination is likely to involve asking epistemic questions *(1) – (6)* about the system to be replaced, and then deciding how the answers to those questions will be used to establish specific safety requirements for the new system. For replacement systems that are intended to provide additional safety (for example, anti-lock braking systems), similar questions must be asked and answered.

(11) *What are the safety implications during transition?* Putting a replacement system into operation requires some form of transition from the old system to the new one. In some cases this transition might be done immediately, but in most cases, a period of time will be required in which both the old and new systems are operating. Regardless of the transition time required for putting the replacement system in place, there will necessarily be some time required for the operators to adjust to using the new system. Understanding the safety of this situation is essential; otherwise, the possibility exists that the risk of accidents during transition may be unacceptably high. Research to develop methods and tools to reason about transition effects could make an important contribution to system safety.

New system questions: At least three epistemic questions are important to answer when developing a truly new system.

(12) *How is the desired level of safety to be determined?* All of the questions mentioned above about existing systems presuppose that a determination has already been made about the level of safety that the system is intended to provide. Determining that level is an essential activity in creating a new system; however,, it is rarely a simple task. Because creating meaningful measures for the absence of something (in this case, unacceptable accidents or losses) is so difficult, proxies for the level of safety are often used instead. Perhaps the most common such proxy involves establishing a permitted probability of failure.

(13) *What can be learned from existing systems?* Even when a new system is not replacing an existing one, its intended functions and use may share some characteristics with existing ones. The knowledge available about any such systems, particularly the knowledge derived from answers to questions (1) – (9), may be important.

(14) *How will novel technologies affect safety?* If any novel technologies will be used in the system, then the effect these technologies may have on safety must be carefully considered. This consideration should include not only the potential direct effects, but also indirect effects, such as how resource matters related to novel technologies may affect the resources available for safety assessment and assurance. Answering the question is complicated by the difficulty of maintaining an accurate knowledge of the impact on safety when operators and users continually adopt to the introduction of new technologies in ways that may not have been anticipated. For example, the introduction of anti-lock braking systems seems to have led to at least some drivers to increase their speeds, rather than continue to drive at the speeds they did before their vehicles had the system, under the belief that the braking system will enable them to stop safely at the faster speeds. Theories of risk homeostasis attempt to explain this behavior (Wilde, 2001), but knowing how to apply those theories in the safety analysis of a specific system is difficult. Research in this area, along with research specific to the novel technologies themselves (for example, research about safety analysis methods for non-deterministic, cooperating, autonomous systems), is needed to improve our ability to answer this epistemic question.

In considering answers to questions (13) and (14), it is important to recognize that novelty can sometimes be disguised as simple extensions of existing approaches. 'The history of engineering is full of examples of dramatic failures that were once considered confident extrapolations of successful designs' (Petroski, 1994).

Common questions: Many important epistemic questions about future systems are common, whether the system replaces an existing one, or is a truly new one. There are at least seven[6] such questions.

(15) *What level of confidence in safety is required?* Once the desired level of safety is determined, the level of confidence that this safety will be achieved must be determined. That is, how certain must the system developers (and regulators if the system being developed requires regulation) be that the system is safe? Complete certainty is not possible. Perhaps a level of confidence analogous to the legal standard 'beyond a reasonable doubt' may be

[6] The original list had ten; we moved one of them to the 'accident-independent questions' section, and combined four into two.

appropriate (Caseley and White, 2009). If not, some other level must be determined. Answering this question may complicated by several factors: disagreements among developers and regulators about what is required; unwarranted confidence on the part of some about the level of safety that can be legitimately achieved; and possible lack of solid evidence in which to establish a confidence level.

(16) *How is knowledge obtained about the intended operational environment?* For any new system, understanding the environment in which it will be used is necessary for determining safety requirements. The difficulty of obtaining the needed knowledge may range from fairly simple to quite hard depending on the specific system. For example, in healthcare, there are clear ethical issues in testing out new devices or procedures on patients, but without such tests the knowledge of the actual operational environment may be limited. For space applications, reproducing the rigors of the eventual operating environment prior to launch is extremely difficult, if not impossible.

(17) *How is the sufficiency of safety requirements assured?* Requirements validation—determining that the requirements completely specify the desired attributes of a system—is well-known to be exceedingly difficult in general. Safety requirements validation is no exception, and because of the potential consequences of incomplete safety requirements, accomplishing it is critical. System developers (and regulators in domains in which regulators play a part) must know, to a sufficient level of confidence, that the requirements developed for the system are sufficient to ensure the necessary level of safety within the intended operational environment of the system. If this is not the case, then there is a danger that subsequent verification will fail to test or analyze satisfaction of requirements that are must be satisfied during eventual operation. Requirements validation research has long been an active area, and will continue to be until methods and tools are developed in which high confidence can be placed.

(18) *How is the sufficiency of implementation assured?* Sufficiency of requirements is not enough to ensure safety. Developers (and, if relevant, regulators) must know, to a sufficient level of confidence, that an implementation created to satisfy these requirements does so in such a way as to preserve the safety inherent in the requirements. This area also has been an active area of research for a long time, and will continue to be so.

(19) *How are assumptions and implications understood?* Recognizing that all requirements and implementations include certain assumptions, developers (and regulators, if relevant) need to know that these assumptions, and the implications of them, are sufficiently understood so that the operational use of the system conforms to them.

(20) *What level of confidence is provided by assessment methods and tools?* The answers to many of the foundational epistemic questions discussed so far will be obtained, at least in part, through the use of various methods and tools. Thus, the level of confidence that can be attached to those answers will depend, at least in part, on the level of confidence that can be legitimately derived from the results obtained from these various methods and tools. For example, how does a formal proof of correctness of a model of a part of the system contribute to the level of confidence compared to extensive testing of a completed system? Because other disciplines (such as science, law, and philosophy) confront a similar epistemic question, studying how those disciplines answer the question seems like a fruitful area of research (Haack, 2007; Holloway, 1995; Holloway, 2002; Toulmin, 2001; Toulmin, 2003; Walton, 1997).

(21) *What is the appropriate level of confidence to be attached to the satisfaction of standards?* This is another question around which much current debate revolves. 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 (Australian Government, 2008a; Australian Government, 2008b; Ministry of Defence, 2007; RTCA/EUROCAE, 1992; Software Engineering Institute, 2002). These differences suggest strongly that a general consensus will not be reached in the near future. Thus, this question needs to be asked specifically for each new system. An important related question that should also be asked is: ‘What precautions are necessary to ensure that evaluations of safety are not biased towards simply trying to convince a regulator that the system is safe enough to be deployed?’

As was true for the questions in the previous sections, some of the questions listed above have been considered in various ways (Brooks, 1987; Hawkins and Kelly, 2000; Jackson et al. 2007; Kelly, 1998; McDermid et al. 2005; Weaver, 2003), but no systematic, detailed research efforts exist for developing cogent, comprehensive answers to all of them, or for ensuring that all the relevant questions are enumerated.

Concluding Remarks

Although few system safety practitioners or researchers may be inclined to say it quite this way, 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. In a previous paper, we enumerated an initial set of foundational epistemic questions. In this paper, we have refined those previously developed questions, discussed some difficulties that exist in answering the questions today, and speculated on possible research that may enable improved confidence in the accuracy of answers to the questions in the future.

References

1. Australian Government (2008a). DEF(AUST)5679 / Issue 2. Safety Engineering for Defence Systems.
2. Australian Government (2008b) DEF(AUST)10679 / Issue 1, Guidance Material for DEF(AUST)5679 / Issue 2.
3. Australian Transport Safety Bureau (2007). *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.
4. Australian Transport Safety Bureau (2008). *Analysis, Causality and Proof in Safety Investigations*. Aviation Research and Analysis Report - AR-2007-053.
5. Bahnsen, G. (1978). "A Conditional Resolution of the Apparent Paradox of Self-Deception", Ph.D. dissertation, University of Southern California.
6. Barlay, S. (1970). *The Search for Air Safety: An International Documentary Report on the Investigation of Commercial Aviation Accidents*. William Morrow & Company, Inc.
7. Brooks, F. P. (1987). "No Silver Bullet: Essence and Accidents of Software Engineering", *IEEE Computer*, **20**, no. 4, pp. 10-19.
8. Caseley, P.R. and T.A.D. White. (2009). "The MOD Procurement Guidance on Software Safety Assurance – Assessing and Understanding Software Evidence", *Proceedings of the IET 4th International Conference on System Safety 2009*.
9. Clark, Gordon H. (1989). *Thales to Dewey*. Trinity Foundation.
10. Collins, J., N. Hall, and L.A. Paul, editors (2004). *Causation and Counterfactuals*. MIT Press.
11. Damar, T. E. (2005). *Attacking Faulty Reasoning: A Practical Guide to Fallacy-Free Arguments*. 5th edition. Thomson-Wadsworth.
12. Ministry of Defence (2007). Defence Standard 00-56, "Safety Management Requirements for Defence Systems", Parts 1 and 2, Issue 4.
13. Greenwell, W. S. (2007). "Pandora: An Approach to Analyzing Safety-Related Digital-System Failures", Ph.D. thesis, School of Engineering and Applied Sciences, University of Virginia.
14. Grudy, T. (2005). *A Practical Study of Argument*. 6th edition, Thomson/Wadsworth.
15. Haack, S. (2007). *Defending Science — within reason*. Prometheus Books.
16. Haddon-Cave, C. (2009). *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. Ordered by the House of Commons to be printed 28th October 2009.
17. Haidt, J. (2006). *The Happiness Hypothesis: Finding Modern Truth in Ancient Wisdom*. Basic Books.
18. Hawkins, R. D. and T. P. Kelly (2000). "A Systematic Approach for Developing Software Safety Arguments", *Proceedings of the 27th International System Safety Conference*, Huntsville, Alabama.
19. Holloway, C. M. (1995). "Software Engineering and Epistemology", *Software Engineering Notes*, **20**, No. 2.
20. Holloway, C. M. (1999). "From Bridges and Rockets, Lessons for Software Systems", *Proceedings of the 17th International System Safety Conference*, pp. 598-607.
21. Holloway, C. M. (2002). "Issues in Software Safety: Polly Ann Smith Co. v. Ned I. Ludd", *Proceedings of the 20th International System Safety Conference*.
22. Holloway, C. M., C. W. Johnson (2006). "Why System Safety Professionals Should Read Accident Reports", *The IET 1st International Conference on System Safety*, London.
23. Holloway, C. M., C. W. Johnson (2009). "Towards a Comprehensive Consideration of Epistemic Questions in Software System Safety", *Proceedings of the IET 4th International Conference on System Safety 2009*.
24. Jackson, D., M. Thomas, L. I. Millett, editors (2007). *Software for Dependable Systems: Sufficient Evidence?* National Research Council, Committee on Certifiably Dependable Software Systems.

25. Jet Propulsion Laboratory (2000). JPL Special Review Board. “Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions”, JPL D-18709.
26. Johnson, C. W. (1997). “The Epistemics of Accidents”, *Journal of Human Computer Systems*, **47**.
27. Johnson, C. W. (2003). *Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting*. University of Glasgow Press, Glasgow, Scotland, United Kingdom. Available on-line at: http://www.dcs.gla.ac.uk/~johnson/book [accessed May 17, 2010].
28. Johnson, C. W. and C. M. Holloway (2007). “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.
29. Johnson, C. W., A Herd, and M. Wolff (2010). “The Application of Resilience Engineering to Human Space Flight”, *Proceedings of the International Association for the Advancement of Space Safety*.
30. Kelly, T. P. (1998). Arguing Safety - A Systematic Approach to Safety Case Management, PhD thesis, Department of Computer Science, The University of York, United Kingdom.
31. Leveson, N. G. (1994). “High Pressure Steam Engines and Computer Software”, *IEEE Computer*, **27**, no. 10, pp. 65-73.
32. Leveson, N. G. (1995). *Safeware: System Safety and Computers*. Addison-Wesley.
33. McCarthy, N. (2008). “Philosophy and engineering”, *Interdisciplinary Science Reviews*, **33**, No. 3.
34. McDermid, J., T. P. Kelly, R. Weaver (2005). “Goal-Based Safety Standards: Opportunities and Challenges”, Proceedings of the 23rd International System Safety Conference, San Diego, California.
35. Oxford University Press (1989). *The Oxford English Dictionary*. 2nd ed. (1989). *Oxford English Dictionary Online*. <http://dictionary.oed.com/>. [accessed May 17, 2010].
36. Petroski, H. (1992). *To Engineer is Human: The Role of Failure in Successful Design.* Vintage Books.
37. Petroski, H. (1994). *Design Paradigms: Case Histories of Error and Judgement in Engineering.* Cambridge University Press.
38. Petroski, H. (1995). *Engineers of Dreams: Great Bridge Builders and the Spanning of America*. Alfred A. Knopf, Inc.
39. Poel, I. and D. Goldberg, editors (2009). *Philosophy and Engineering: An Emerging Agenda*. Springer.
40. RTCA/EUROCAE, (1992). DO-178B/ED-12B, “Software Considerations in Airborne Systems and Equipment Certification”.
41. Safety Critical Mailing List Archive (2010). Available at http://www.cs.york.ac.uk/hise/safety-critical-archive/. [accessed multiple times between January 1, 2010 and May 20, 2010].
42. Snook, S. A. (2000). *Friendly Fire: The Accidental Shootdown of U.S. Black Hawks over Northern Iraq.* Princeton University Press.
43. Society of Automotive Engineers (1996a). *Certification Considerations for Highly-Integrated or Complex Aircraft Systems.* SAE ARP 4754.
44. Society of Automotive Engineers (1996b). *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. SAE ARP 4761.
45. Software Engineering Institute (2002). *CMMI^{SM} for Software Engineering (CMM-SW, V1.1)*. CMU/SEI-2002-TR-029.
46. Toulmin, S. E. (2001). *Return to Reason*. Harvard University Press.
47. Toulmin, S. E. (2003). *The Uses of Argument.* updated edition, Cambridge University Press.
48. Vaughan, D. (1996). *The Challenger Launch Decision*. The University of Chicago Press.
49. Walton, D. (1997). *Appeal to Expert Opinion*. The Pennsylvania State Press.
50. Weaver, R. A. (2003). The Safety of Software - Constructing and Assuring Arguments. PhD thesis, Department of Computer Science, The University of York.
51. Wilde, G. J. S. (2001). *Target Risk 2: A new psychology of safety and health*.

Biography

C. Michael Holloway, NASA Langley Research Center, 100 NASA Road, Hampton, VA 23681-2199, telephone - (757) 864.1701, facsimile - (757) 864.4234, e-mail - c.m.holloway@nasa.gov.

C. Michael Holloway is a senior research engineer at NASA Langley Research Center. His primary professional interests involve epistemic issues in system safety and accident analysis for software intensive systems. His primary real interests include theology, education, constitutional law, baseball, volleyball, and roller coasters. He is a member of the IEEE, the IEEE Computer Society, and the International System Safety Society.

Chris.W. Johnson, DPhil, MA, MSc, FBCS, CEng, CITP, Department of Computing Science, University of Glasgow, Glasgow, G12 8RZ, Scotland, UK, telephone +44 (141) 330 6053, facsimile +44 (141) 330 4913, e-mail – Johnson@dcs.gla.ac.uk, web page http://www.dcs.gla.ac.uk/~johnson

Chris Johnson is Professor of Computing Science at the University of Glasgow in Scotland. He heads a small research group devoted to improving the reporting and analysis of incidents and accidents across safety-critical domains ranging from healthcare, to the military to aviation and rail.

Appendix

The list below collects the foundational epistemic questions discussed in this paper in one place.

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 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 level of confidence is provided by assessment methods and tools?
(21) What is the appropriate level of confidence to be attached to the satisfaction of standards?