TOWARDS A COMPREHENSIVE CONSIDERATION OF EPISTEMIC QUESTIONS IN SOFTWARE SYSTEM SAFETY

C. M. Holloway*, C. W. Johnson†

* NASA Langley Research Center, 100 NASA Road, Hampton VA 23681, USA, c.m.holloway@nasa.gov
†Dept. of Computing Science, University of Glasgow, Glasgow G12 9QQ, UK, johnson@cs.gla.ac.uk

Keywords: epistemology, safety, software, confidence.

Abstract

For any software system upon which lives depend, the most important question one can ask about it is, 'How do we know the system is safe?' Despite the critical importance of this question, no widely accepted, generally applicable answer exists. Instead, debate continues to rage over the question, with theorists and practitioners quarrelling with each other and amongst themselves. This paper suggests a possible way forward towards quelling the quarrels, based on refining the critical safety question into additional questions, which may be more likely to have answers on which a consensus can be reached.

1 Introduction

'Is the system safe?'

'Do we think 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 *think* the system is safe, then it *is* safe; and if we do not think 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 think a system is safe when it is not, and we may think a system is not safe when it is.

The orthogonality of 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.

We believe that to understand adequately the discrepancy between current practice and theory, and to speculate intelligently about what may happen in the future, foundational epistemic questions related to software system safety must be carefully and systematically considered. In this brief paper, we suggest what some of those fundamental questions may be.

2 Definitions

We begin with definitions for, and discussion about, some of the words and phrases that are used in the paper. Although some of the questions listed later in the paper can be understood without understanding these definitions, some cannot.

2.1 Concerning Knowledge

Epistemic is an adjective meaning 'of or relating to knowledge or degree of acceptance' [25]. *Epistemology* is a noun defined as 'the theory or science of the method or grounds of knowledge' [25]. Epistemology is one of the major branches of study in philosophy [5]; 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].

[1] Although we *know* that any philosopher reading this paper will consider the discussion in this section woefully simplistic and

Regardless of a particular individual's use of the three verbs, he or she may be wrong. For example, someone may believe, think, or know that the 4th International Conference on System Safety 2009 is being held in Birmingham. The strength of the individual's level of confidence does change the fact that he or she is simply wrong [3, 7, 11].

2.2 Concerning Safety

The noun *safety* can 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 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 acceptable.

Understanding the practical definition of safety thus requires understanding the meaning of *acceptable*. How much freedom from accidents and losses is acceptable? Answers to that question have varied over time, among different domains, and even among different individuals [22, 27, 33].

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 one of these facts alone necessarily implies acceptable safety, but taken together they do.

2.3 Understanding the Original Questions

Based on the above definitions, the first question that opened this paper ('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 think the system is safe?') may be better stated as 'Do we know the system is safe?' That is, '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 revert to the shorter form, relying on the reader to mentally translate to the longer form when necessary.

incomplete, we *believe* that it is sufficiently detailed and complete for the intended audience of the paper.

3 Foundational Epistemic Questions

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 wide range of existing opinions, and the emotional fervour with which these opinions are held [28], 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 the remainder of this section, we suggest what some of the foundational questions may be. These suggestions are not complete. Not only are there additional questions that should be considered, but most of the questions listed below need to be further refined. Questions about existing systems are discussed first, followed by questions about future systems.

3.1 Questions About Existing Systems

Existing systems may be divided into two main categories: systems that have been operating for sufficiently long that they are known to be safe, and systems that have not been operating that long[2]. Epistemic questions concerning both categories are generally similar, with the exception of the following question:

- What is necessary for a system in use to be considered to have its safety effectively demonstrated? Is passage of some period of time without any unacceptable accidents or losses sufficient? Or is something additional needed?

- What is known about the effect of the specific operational environment on the safety of the system? Specifically, is that effect known well enough to be able to accurately assess the safety consequences of changes in the operational environment?

The questions that apply to both categories include the following:

[2] A third category is also possible: systems that are operating and considered to be unsafe. For the purposes of this paper, we assume, idealistically, that such systems are taken out of service as soon as they are recognized as unsafe.

- How is operational safety best measured? That is, what information must be collected and analyzed to provide adequate confidence that a system in use is truly acceptably free from accidents and loses?

- How should differences in evaluations of safety 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?

- To what extent should measures of operational safety be compared to pre-deployment evaluations of expected safety? Might regular comparisons result in better understanding of the efficacy of system safety evaluation procedures and tools?

- What maintenance, if any, does the system require to maintain its safety? What information must be collected to ensure adequate maintenance is performed?

When an accident or loss occurs in an existing system, additional epistemic questions arise, including the following:

- 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? What do investigator do if adequate information is not available? (See [18] for example of such a situation).

- How do the investigators know that they have found the relevant causes and contributing factors to the accident or loss?

- How can the knowledge gained from identifying the causes and contributing factors be used to improve the safety of the existing system?

- How can the knowledge gained from identifying the causes and contributing factors be transferred to those responsible for similar existing systems and designers of similar future systems?

- How can the knowledge gained from identifying the causes and contributing factors be collected and maintained so that it is available in an understandable form for as long as it may be relevant?

- What can be done to encourage designers and engineers to make use of the available knowledge?

Some of the questions listed above have been considered in various ways (see for example [6, 10, 15, 19, 20, 22, 23, 26, 27]), but we are unaware of any systematic, detailed research efforts aimed towards developing methods for providing cogent, comprehensive answers to all of them. Nor are we aware of any efforts towards fully enumerating all of the relevant epistemic questions that should be answered.

3.2 Questions About Future Systems

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.

Epistemic questions relevant to both categories of future systems include the following:

- What level of confidence in the safety of the system is required? That is, how sure must the system developers (and regulators if the system being developed requires regulation) be that the system is safe? Is a standard analogous to 'beyond a reasonable doubt' strong enough? Or should the standard be even stronger?

- How do system developers obtain adequate knowledge about the intended operational environment for the system?

- How do system developers know that the requirements developed for the system are sufficient to ensure safety within the intended operational environment of the system?

- If sufficient requirements are developed, how do developers know that a design created to satisfy these requirements does so in such a way as to preserve the safety inherent in the requirements?

- Given safety-ensuring requirements and a safety-preserving design, how do developers (and regulators in domains in which regulators play a part) know that the implementation of the design results in a safe system?

- What level of confidence can be legitimately derived from the results of various methods and tools for assessing the system? 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? What can be learned from other disciplines that might help to answer questions such as this [12, 14, 16, 30, 31]?

- Recognizing that all requirements, designs, and implementations include certain assumptions, how do developers (and regulators) know that these assumptions, and the implications of them, are sufficiently understood

so that the operational use of the system will conform to them?

- What is the appropriate level of confidence to be attached to the satisfaction of standards? This is one of the questions 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 [1, 2, 8, 9, 29].

- 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?

- When changes are made to an operational system, what knowledge is required to ensure that those changes do not adversely affect the safety of the system, and how is that knowledge analyzed to insure that safety is preserved?

Epistemic questions specific to truly new systems include the following:

- How is the appropriate level of safety for the system to be established?

- Is knowledge available from any existing system that may be helpful in developing the requirements for the new system?

- Are any novel technologies going to be used in the system? If so, how will the safety aspects of those new technologies be assessed? In considering these questions, it is important to recognize that novelty can sometimes be disguised as simple extensions of existing approaches. As Petroski wrote, 'The history of engineering is full of examples of dramatic failures that were once considered confident extrapolations of successful designs' [26].

Finally, epistemic questions specific to systems that are created to replace already existing systems include the following:

- Assuming the new system is intended to be 'at least as safe as' the existing system, how is that baseline to be established?

- What knowledge about the existing operational system is required to permit the baseline to be established?

- How is that baseline to be used in evaluating the expected safety of the new system?

- What are the potential safety implications of the transition from the existing system to the new one? How long will this transition take? How much can be known about the safety of the combined systems during the transition period?

As was true for the questions in the previous section, some of the questions listed above have been considered in various ways [4, 13, 17, 21, 22, 24, 32], 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.

4 Concluding Remarks

This paper has presented an initial attempt to enumerate a set of foundational epistemic questions concerning software system safety. We recognize that this set is incomplete, and thus are keen to receive comments on these questions from the conference participants, and plan to revise and expand the set of questions based on those comments. Potential future work beyond revision and expansion includes organizing the questions into a useful taxonomy, explaining how existing software safety approaches and tools contribute to answering the questions, and speculating about the future research that is needed to develop a complete and coherent set of questions and answers.

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

[4] F. P. Brooks. "No Silver Bullet: Essence and Accidents of Software Engineering", *IEEE Computer*, **20**, no. 4, pp. 10-19, (1987).

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

[6] J. Collins, N. Hall, and L.A. Paul (eds.). *Causation and Counterfactuals*. MIT Press, (2004).

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

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

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

[10] 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).

[11] T. Grudy. *A Practical Study of Argument.* 6th edition, Thomson/Wadsworth, (2005).

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

[13] 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).

[14] C. M. Holloway. "Software Engineering and Epistemology", *Software Engineering Notes*, **20**, No. 2, (1995).

[15] C. M. Holloway. "From Bridges and Rockets, Lessons for Software Systems", *Proceedings of the 17th International System Safety Conference*, pp. 598-607 (1999).

[16] C. M. Holloway. "Issues in Software Safety: Polly Ann Smith Co. v. Ned I. Ludd", *Proceedings of the 20th International System Safety Conference*, (2002).

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

[18] 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).

[19] Johnson, C. W. "The Epistemics of Accidents", *Journal of Human Computer Systems*, **47**, (1997).

[20] Johnson, C.W. *Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting.* University of Glasgow Press, Glasgow, Scotland, United Kingdom, (2003). Available on-line at: http://www.dcs.gla.ac.uk/~johnson/book [accessed July 6, 2009]

[21] Kelly, T. P. Arguing Safety - A Systematic Approach to Safety Case Management, PhD thesis, Department of Computer Science, The University of York, United Kingdom (1998).

[22] Leveson, N.G. "High Pressure Steam Engines and Computer Software", *IEEE Computer*, **27**, no. 10, pp. 65-73, (1994).

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

[24] McDermid, J., Kelly, T. P., Weaver, R. "Goal-Based Safety Standards: Opportunities and Challenges", Proceedings of the 23rd International System Safety Conference, San Diego, California, (2005).

[25] *The Oxford English Dictionary.* 2nd ed. (1989). *Oxford English Dictionary Online*, Oxford University Press. <http://dictionary.oed.com/>. [accessed July 6, 2009]

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

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

[28] Safety Critical Mailing List Archive. Available at http://www.cs.york.ac.uk/hise/safety-critical-archive/2009/ (2009). [accessed multiple times between June 15, 2009 and September 30, 2009]

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

[30] Toulmin, S. E. *The Uses of Argument.* updated edition, Cambridge University Press, (2003).

[31] Walton, D. *Appeal to Expert Opinion.* The Pennsylvania State Press, (1997).

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

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