Carl Sandom* and Robert D Macredie**

**Systems Development Centre*

Royal Air Force Bentley Priory

Stanmore, Middlesex, HA7 3HH

Tel: +44 181 8387314; Fax: +44 181 8387539

***Department of Information Systems and Computing,*
Brunel University, Uxbridge, Middlesex, UB8 3PH
Tel: +44 1895 203374; Fax: +44 1895 203391

E-mail: Carl.Sandom@brunel.ac.uk; Robert.Macredie@brunel.ac.uk

SOFTWARE HAZARDS AND SAFETY-CRITICAL INFORMATION SYSTEMS

What is a 'Software Hazard'?

A common mantra in the Safety-Critical Systems arena is that there is "no such thing as a software hazard". One justification for this view is the belief that software systems cannot cause harm directly. Instead, hazards are only related to physical artefacts, such as computer hardware. Yet, others have suggested that safety-critical software is considered to be a major contributor to risk in many computer-based systems (Storey 1996;Leveson 1995) - and it is only a short step between risk and the realisation of risk as hazard. The often contradictory positions are made more acute in systems where the risks are difficult to assess and hard to quantify. A prime example of this is in computer-based information systems, where the interaction between user and system may be the area of highest potential risk. User actions may cause systems to respond in the anticipated way, but there may be unforeseen consequences. A well-documented example of this is the failure of the London Ambulance Service Computer-Aided Dispatch (LASCAD) system in October 1992. It has been suggested that this caused significant delays in the dispatch of ambulances which may have contributed to between 20-30 deaths, yet the developers have argued that the system behaved as it was meant to, and that the fault was not in the software (Benyon-Davies 1995).

The argument about whether or not LASCAD or other software systems are safety-critical - and if they are where the hazards lie - is one which we would contend is dependent on your view of the system. Our view is that the systems boundary encompasses not only software and hardware, but - just as importantly - people (O'Brien 1996; Avison and Fitzgerald 1995). If any component of the system can either directly or indirectly affect safety, there is an argument that the system should be considered safety-critical. In this article we will try to shed light on the extent to which it is only the placement of the (imaginary) system boundary that differentiates information systems like LASCAD from traditional safety-critical systems and the implication this has for our view of where system hazards lie.

What are Safety Critical Systems?

The integrity of modern technological systems is a growing social concern, particularly as many computer-based systems are now being developed with the potential for increasingly catastrophic consequences from a single accident. Computer systems that control everyday activities from power generation to traffic control may have the potential to contribute to - if not to cause - accidents on a large scale. At one end of the scale are large industrial accidents such as the Union Carbide chemical plant in Bopal, India, which killed 2,000 to 3,000 people and seriously injured over 200,000 (Ayres and Rohatgi 1987); at the other may be the LASCAD example. There may be variations in scale and the extent to which the problem can be definitively shown to be a software fault, but the public perception of such problems tends to be that a computer system caused them by 'going wrong'.

This may not sit well with classical definitions of a safety-critical system as one which can exert a direct influence on the safety of its users and on the safety of the general public. Using this argument,

the existence of unsafe drivers could be disputed as it is the vehicle that directly causes harm and not the driver (Storey 1996). In our examples, it may well have been the use - or misuse - of the system which directly contributed to the accidents rather than the system causing a problem directly, supporting the view that they did not offer a software hazard. We would, however, argue that before pursuing this line of argument you have to carefully consider what constitutes the system and where the system boundary is drawn.

A Developing View of Safety-Critical Systems

Any attempt to re-consider the safety-critical system boundary as one which begins to encompass users and even aspects of the organisation in which the system operates is bound to be problematic. Such moves bring unprecedented levels of complexity: the focus of the systems and their use become socio-technical; they tend to be inherently unpredictable; and they are impossible to formalise.

There is little wonder, then, that such moves are likely to be resisted, with safety remaining the province of only physical artefacts, such as hardware, rather than also being associated with intangible software products. This view suggests that any safety problems arising through information systems, for example, can only be indirect and further implies that the term 'software hazard' is erroneous and should be taken to mean a software failure mode that can contribute to a hazard.

Whilst this perspective may simplify, or restrict, our view of safety-critical systems, it adds very little to our understanding of the issues involved since it marginalises problems we know to exist. For example, a narrow view of safety-critical systems would probably deny any safety issues in stock-control systems, but if such a system fails to provide the correct information, or the information is interpreted incorrectly by its user, and an incorrect component is fitted to an aircraft as a result, the consequences may be lethal.

In our view, the distinction between the physical and the intangible - and therefore whether or not systems offer software hazards - is largely irrelevant when considering the safety of systems. Rather, it is a question of where the system boundary is perceived to exist. Drawing a broader system boundary than is traditionally associated with safety-critical systems leads us to more challenging views of safety and the role of systems in contributing to safety issues. Information systems, for example, often provide organisational and environmental data to inform decision making. Depending on its use, this can lead to erroneous actions affecting safety as was the case when the crew of the USS Vincennes incorrectly interpreted the information presented by their system and a decision was taken to shoot down a commercial airliner killing 290 passengers (Greatorex and Buck 1995). Yet, such information systems are not generally considered to be safety-critical.

Taking a broader view implies that if system safety is to be enhanced then the design and operation of all the components which can directly or indirectly affect safety should be considered. To pursue this goal, there is a need for collaboration between different disciplines, such as traditional safety-critical systems, information systems, human-computer interaction, and organisational behaviour. In certain situations, where systems involve interaction with users in complex social and organisational settings, it is important that we critically re-consider what constitutes the system and where the systems boundary is to lie. For us, taking this wider, systemic view brings a definition of what constitutes a safety-critical system sharply into focus. If a system failure can initiate an "unintended sequence of events that causes death, injury, environmental or material damage" (MoD 1996, pp. A-1) these systems may be classified as safety-critical, irrespective of how or where that failure arises.

Understanding the system failure has to remain the goal. As systems become increasingly socio-technical and embedded in more and more complex organisational contexts, only a wider view of what constitutes a safety-critical system will bring us closer to understanding such failures. This also encourages us to re-consider the notion of 'software hazard'. Ultimately, the phrase loses importance since our concern should be system hazards, with the system boundary being cast broadly to encompass all relevant system components. Attaching the hazard to one component over-simplifies the situation and may lead us to ignore relevant issues.

References

Avison D E and Fitzgerald G (1995), *Information Systems Development: Methodologies, Techniques and Tools*, London, McGraw-Hill.

Ayres R U and Rohatgi P K (1987), *Bhopal: Lessons for Technological Decision-Makers*, Technology in Society, 9 pp.19-45.

Benyon-Davies P (1995), *Information Systems Failure: The Case of the London Ambulance Service's Computer Aided Dispatch Project*, European Journal of Information Systems, 4 pp.171-184.

Greatorex G L and Buck B C (1995), *Human Factors and Systems Design*, GEC Review, 10 (3) pp.176-185.

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

MoD Def Stan 00-56 (1996), *Safety Management Requirements for Defence Systems, Part 1: Requirements*, December 1996.

O'Brien J A (1996), *Management Information Systems: Managing Information Technology in the Networked Enterprise*, London, Irwin.

Storey N (1996), *Safety-Critical Computer Systems*, London, Addison-Wesley.