PJC/13/06/1997
AVN-97/007

Safety, Reliability and Software Based System Requirements [1]

P.-J. Courtois AVN, Brussels

When discussing the use of computers and software for safety-critical functions, it is useful to clarify the distinction between reliability and safety. One is typically be interested in the safety of an overall system, i.e. a nuclear plant, which is comprised of interacting lower level systems, such as instrumentation systems. These lower level systems are themselves comprised of lower level systems, and so on. Safety-critical computer systems form themselves a part of the overall safety system of a nuclear plant. Much of the concern of this note, in turn, is centred upon the role of software in such a computer system.

Safety is the attribute of a system - e.g. a nuclear power plant - to be free from the occurrence of accidents, i.e. from the undesired events that lead to catastrophic consequences such as health and environmental effects of radiation and radioactive contamination. Safety is achieved through the use of *reliable* structures, components, systems and procedures. *Reliability* is the probability that a system or component will perform its intended function for a prescribed time and under stipulated environmental conditions.

Reliability may thus be determined by the probability of failure per demand, whilst safety is also determined by the consequences of these failures. In a reactor safety system, for instance, the primary functionality concerns the requirement to shut the reactor down safely when needed, and keep it in a safe state for a specified period of time. If the software in a safety system is unreliable, i.e. if there is a too-high probability of its not carrying out this shut-down function correctly when demanded, then there will be an unacceptable effect upon the safety of the wider system. If a computer-controlled control panel prioritises or filters alarm signals incorrectly, there can also be an adverse effect on safety.

The achievement of the required reliability by the hardware and software alone, however, is not enough to guarantee overall plant safety. If the specification of the safety-critical system is inadequate then the overall nuclear system may be unsafe even though the hardware and software implementation of the safety-system is completely reliable (with respect to its specification). Moreover, in a more general context, the events leading to an accident are almost never limited to a computer failure, but are a complex combination of equipment failures, faulty maintenance, human actions and design errors. Some accidents even result from a sequence of events, none of which may involve a component failure. Each component may reliably work as specified, but together they may create a hazardous system state.

[1] Contribution to the Report of the Advisory Committee on the Safety of Nuclear Installations Study Group on Safety of Operational Computer Systems.

An example that reliability alone is not enough to guarantee safety is given in Nancy Leveson's book "Safeware" and is provided by domestic fuses. Their failure frequency is estimated at 10^{-6} or 10^{-7} per year. Fuses, however, can be wrongly calibrated or replaced by copper wire. The frequency of these errors has been estimated at 10^{-3} per year. Further improvements of the reliability of fuses would therefore never make their use safer.

On the other hand, in nuclear power plants as elsewhere, there are also sources of unreliability which would not be regarded as contributing to the overall plant risk - for example, component failures that can be proved to lead to a safe state. For instance, the protection relays (nowadays often microprocessor based), which protect the power supplies of safety injection mechanisms against transient peaks of voltage or current, may not necessarily be considered as impacting safety if they fail to trip on demand. Depending on their role at the plant level, their failure to trip may or may not leave the plant in a safe state. Likewise, the reliability contribution required from a diverse system may be much lower than that of the primary safety system or line of defence. An example is given by the 'off the shelf' computer based radiation detectors which are sometimes used as an ultimate detection mechanism for containment isolation in the event of the protection system failing to detect a loss of coolant accident (LOCA).

Thus, it is important when discussing reliability and safety to have in mind both the system (or sub-system) of interest *and* its environment (often a wider system). In fact in the nuclear engineering community, it is normal to reserve the word *safety* for use as a property of an overall nuclear system, and to refer only to the *reliability* of any computer systems involved as well as to the adequacy of its set of requirements that have been identified and specified. With this usage - that we adopt in this report - unreliability is associated with undesired departures from the specified behaviour. In contrast, breaches of safety are associated with unexpected and undesired behaviours that had not been specified or inadequately specified.

A natural consequence of these considerations is that solutions to safety issues must start with system, rather than software engineering. Clearly, the identification of the possible events that are to be regarded as important to safety is a key part of the determination of the safety requirements of, e.g. a nuclear protection system. This is a complex task which belongs to the world of nuclear safety engineers. They have to anticipate all possible failure modes of the protection system, define its functionality, and non-functional behaviour (e.g. internal monitoring), and do this whatever technology will be used: hardware, software, hydraulics, etc. The safety requirements are thus translated into functional and non functional requirements of a protection system. It is then decided what technology to use, and hardware and/or software specifications are written. What is required from the implementation is demonstrable satisfaction of its specified reliability.

In other words, the impact of safety on design should ideally be confined to the functional and non functional *specifications* of the system. Software and hardware should then simply be required to be a reliable implementation of these requirements demonstrably correct and tolerant of hardware random faults. In these circumstances, the somewhat ambiguous notion of "software hazard" used by certain authors could be dispensed with.

This is not to say that safety can be ignored within the software implementation team (or any other implementation team). The initial version of the specification will not be perfect - it may have omissions, inconsistencies and requirements which are at least non-optimal, if not incorrect. One of the important tasks of the implementation team is to refine the specification and in so doing they may have to feed back changes to the system team to be included in new updated versions of the specifications. . In addition, where there are design options for the implementation team, the relative effect on safety should form part of the decision making process. So, concern for safety is indispensable within the software team, but the matters of concern and the techniques applicable are different from those of the system engineers who specify the safety requirements.

This concern for safety is a crucial issue and is related to the role of the individual engineers in large projects where software is a component only. At stake is the production of system specifications that are both complete and understandable by all parties involved in the design. The challenge is to develop methods for specifying in the system requirements “everything” concerning safety in a way which is understandable to computer hardware and software designers.

This necessity is confirmed by the return of experience of incidents involving software. For instance, adequate protections against the Therac-25 accidents could and would probably have been integrated into the software if the software designers had been given specifications that included a correct and complete description of the possible failure models of the Therac machine and of the possible hazardous misuses of the operator interface.

Whilst there is a certain element of judgement involved in the identification of possible events that are important to safety, it is usual in those industries in which safety-critical systems are common to have a systematic safety analysis process which includes procedures to identify these events. In some nuclear safety systems, for example, the existence of a safe state makes the problem simpler than for other systems: reliable delivery of appropriate responses to the demands placed upon the safety system is the major safety issue, and the number and nature of these demands can be assumed to be well-understood. In other applications, however, particularly those involving systems that do not have a safe state (e.g. an operator assistance system, or an aircraft flight control) and where safety issues are mainly associated with ensuring that unexpected and undesirable events do *not* happen, their identification and analysis can be much harder, so that the completeness and correctness of the safety analysis procedure should itself be taken into account in the safety evaluation.

It should also be emphasised that safety, like reliability, is not an absolute. We can say that a system is sufficiently safe, but not that it is completely safe. Much of our discussion will centre on the evidence and arguments we need to deploy to make claims for this *sufficiency*. Ideally, such claims will be expressed numerically; thus we might require that the probability of failure upon demand of a safety system - depending on the consequences of the failure - be smaller than some number emanating from the wider plant safety case, which provides the context for the requirements specification of the safety system.

Finally, this distinction between levels at which safety and reliability requirements apply helps to clarify some implications of the ALARP principle (as low as reasonable practical), especially if ALARP requirements must apply to the design of systems of limited demonstrable reliability that have to be used within high risk environments

One possible interpretation of ALARP indeed is that, when feasible, safety at plant level should be improved - i.e. risks decreased - by providing increased reliability to the systems in charge of those safety functions, provided that the effort or costs at which this increased reliability can be achieved and demonstrated are not grossly disproportionate to the safety improvement.

This implies that it is possible to compare *marginal improvements in safety* (marginal risk decreases) with the *marginal costs of the increases in reliability*. Nuclear risks may offer this possibility when they are quantified (i.e. in terms of event probability and of radiation releases), and when the failure rate improvements of the systems controlling the relevant events can be evaluated. This comparison of marginal variations does not in principle require a common measure, but simply that both risk and the marginal cost or efforts to improve reliability can be realistically assessed. This assessment can however be problematic, especially when design faults have to be taken into account.

Another issue that can be raised in practice by the application of the ALARP principle is that one may have to be able to confidently evaluate these potential marginal variations before the detailed design and the implementation of the modifications are actually completed, or even started.
