Can Human Performance be Addressed Within the Current Safety Assessment Process?

Maarten Boasson
Naval Command and Control System
Anti-Weapon Systems
Universiteit van Amsterdam
Hollandse Signaalapparaten B.V.
P.O. Box 42
7550 GD Hengelo, The Netherlands

Can human performance be addressed within any safety assessment process?

Content

- Context
- Questions
- Observations
- Conclusions

System boundary

- Human operator is not part of a system
  - we design systems; it is presumptious to suggest we can design human operators
- The interface through which an operator interacts with a system, is part of that system
  - including rules and constraints for usage

Human performance

- Must be studied in relation to the system
- It has many facets
  - reaction time
  - quality of decision, given available information
  - manipulation of controls
  - alertness
  - bias

  any of these can lead to disaster!
- Can human performance be quantified? Some aspects of human performance can such as response time to a given stimulus, but others escape even formal description.

Safety

- A system is intrinsically safe, if under no circumstance a catastrophe is caused by actions in which the system is involved.
- It seems unlikely that such systems can be built!
  - Relative to the defnition of catastrophe
- Safety of systems relies on three aspects:
  - correctness of the design
  - correctness of the implementation, and
  - operation within the design limits

  under the assumption of correct specifications.

Paper presented at the RTO HFM Workshop on "The Human Factor in System Reliability – Is Human Performance Predictable?", held in Siena, Italy, 1-2 December 1999, and published in RTO MP-032.

Limits of our abilities

- It is impossible to predict all possible circumstances a system can be in.
  - We do not generally control the environment:
    - turbulence
    - hijackers
    - imperial to metric conversion
- It is equally impossible to predict all possible system malfunctions, and the associated system behaviours.
  - At least in software intensive systems.
- It is utterly impossible to foresee all possible human actions.
- Is it possible to define all allowed system states?
- If so, can a system be constrained to always be in one of these states?
- Thus, e.g. can faulty operator action be corrected automatically?
- Currently there is no rigorous way to demonstrate correctness of a design; this is true regardless of the engineering discipline involved (but probably more so for software than for other disciplines).
- Establishing that a design is correct w.r.t. a given specification, is a matter of extensive discussion, walkthroughs, etc. Thus, in essence it is a matter of belief and trust.
- Formal checking of conformance between a (certified) design and its implementation, is beyond our abilities.
- At best, extensive testing suggests there are no major implementation errors. Note that software does not really have an implementation stage: the complete design is in itself the implementation. That, unfortunately, does not make it any easier to demonstrate correctness.
- Operation within the design limits requires absence of malfunctions in all of the parts, as well as correct behaviour of the system operator(s).
- Overload in software systems typically occurs as a result of incorrect functioning of either sensing devices (producing more measurements than anticipated), or operators (issuing illegal commands, e.g.).
- Any process aiming at establishing safety of a system, must necessarily contain a large component that relies on human insight, rather than on formal techniques.
- The resulting qualification can therefore not be construed as a guarantee for safety;
  at best, it provides some measure of confidence that the system is unlikely to fail under circumstances for which it was designed.
- It is questionable whether probabilities given for catastrophic failures of software intensive systems have any useful interpretation.
  - What does the aircraft industry's 10^{-9} mean?
  - The traditional reliability model is unsuitable for software.
- Operator interfaces are part of the design, but operators are not.
- The best that can be done is to specify acceptable operator behaviour under as a wide a variety of circumstances as possible.
  - But we do not know all possible circumstances.
- Providing "natural" interfaces helps
- Operator interfaces can potentially be designed to limit the operator to perform acceptable actions only. This severely reduces the effectiveness of the human operator when the system no longer meets the design constraints. In fact, it reduces the operator to an agent that could have been automated.
  - E.g. an aircraft could be made to refuse execution of an excessively steep dive

- A system can be designed to correct human actions that are considered erroneous (or unsafe), i.e. leading to the system going out of its allowed boundaries. This is similar to refusing illegal commands, but may allow a little more freedom, at greater risk.
- How often can a system recognize such actions? Until we have a formalism for their characterization, it seems difficult for a system designer to make a system sensitive to them.
- Operational procedures and operator training can go a long way in making human behaviour predictable.
- In the limit case, processes developed for assessing system safety, can also be used for human performance w.r.t. safety issues. But then, the operator has been reduced to a finite automaton, and could (should?) have been replaced.
- Operator behaviour not totally governed by operational procedures can hardly be analysed for potential effect on system safety.
- How do we quantify human behaviour?
- The possibility that a human operator will use the interface in unforeseen and dangerous ways, must be taken into account; but how?
  - Note that dangerous situations may well be the result of long chains of interactions between operator and system.
- There is a fundamental conflict between predictability of human behaviour and the ability of man to act intelligently.

- There is a fundamental conflict between predictability of human behaviour and the ability of man to act intelligently.

- Intelligent actions are generally necessary to compensate for system errors (wether due to design faults or material failures).

Conclusion

- For a system to be operated safely, an intelligent human operator is necessary.
- However, a human operator is an intrinsically unsafe component of the <human, system> pair.
- Training and selection are our best friends for improving human performance.
- Quantitative measures for system safety are highly suspect when software is involved.
- Safety assessment can at best give qualitative indications of the likelihood that operators will violate system safety rules.