EXTENDED ABSTRACT

Safety-critical systems are those systems whose failure could result in loss of life, significant property damage, or damage to the environment. There are many well known examples in application areas such as medical devices, aircraft flight control, weapons, and nuclear systems. A safety critical system is a system where human safety is dependent upon the correct operation of the system. The emphasis of this paper is on the software element of safety critical systems, which for convenience is often referred to as safety critical software. However, safety must always be considered with respect to the whole system, including software, computer hardware, other electronic and electrical hardware, mechanical hardware, and operators or users, not just the software element. Safety critical software has been traditionally associated with embedded control systems. Many safety-critical systems rely on software to achieve their purposes. The number of such systems increases as additional capabilities are realized in software. Miniaturization and processing improvements have enabled the spread of safety-critical systems from nuclear and defense applications to domains as diverse as implantable medical devices, traffic control, smart vehicles, and interactive virtual environments. Metrics are commonly used in engineering as measures of the performance of a system for a given attribute. For instance, in the assessment of fault tolerant systems, metrics such as the reliability, and the Mean Time To Failure (MTTF) are well-accepted as a means to quantify the fault tolerant attributes of a system with an associated failure rate. For safety-critical computer systems, safety has higher priority than reliability. Thus, reliability and MTTF are not enough to model and describe safety-critical computer systems. New metrics are needed to quantitatively assess the safety of safety critical computer systems. Quality models like McCall’s, Boehm’s, and ISO 9126 that have been developed to measure the quality of software cannot be used to differentiate the safety critical from the non-critical aspects of critical systems. Because of this limitation, this paper will propose a set of safety metrics that can be used to identify potential safety related problems and deficiencies early in the software development life cycle. The objective of this paper is to identify and propose a set of five metrics that can be used in the safety assessment of safety-critical systems. This paper also identifies the appropriate software development phases in which to collect these safety metrics. A 3-state Markov model is used to model a safety-critical system.

A laboratory prototype safety-critical system viz. a Railroad Crossing Control System (RCCS) is developed in order to validate the safety metrics.

CONCLUSION

This study aims to solve the problem of how to scientifically identify appropriate software safety metrics in each development phase of safety critical computer systems. This paper presents a set of safety metrics that can be used to evaluate the maturity of hazard analysis processes and its interaction with the software development process. Because a software error in a computer controlled critical system can potentially result in death, injury, loss of equipment or property, or environmental harm, a safety metrics set was developed to ensure that the safety requirements are well understood and correctly implemented by the software engineer. A 3-state Markov model is used to model a safety-critical system.

FUTURE WORK

The research work described in this paper is only the initial stage of a long-term study in software safety metrics identification, development and validation. Additional research is on-going, to further validate the safety metrics by applying them to a laboratory prototype safety-critical system viz. a software-based Railroad Crossing Control System (RCCS). Future work to validate this safety metrics set includes further refinement and formal definition of measurements and application of the model to a larger scale project.

REFERENCES