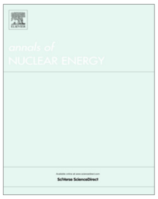
Annals of Nuclear Energy 57 (2013) 294–303

Contents lists available at SciVerse ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene



Sensitivity analysis on the effect of software-induced common cause failure probability in the computer-based reactor trip system unavailability

Shahabeddin Kamyab [a], Mohammadreza Nematollahi [a,b,*], Golnoush Shafiee [a]

[a] School of Engineering, Shiraz University, 71348-51154 Shiraz, Iran
[b] Safety Research Center of Shiraz University, 71348-51154 Shiraz, Iran

ARTICLE INFO

Article history:
Received 9 June 2012
Received in revised form 13 January 2013
Accepted 16 January 2013
Available online 15 March 2013

Keywords:
Reactor trip system
Software common cause failure
Fussell–Vesley
Sensitivity analysis
Risk Achievement Worth

ABSTRACT

The reactor trip system has been digitized in advanced nuclear power plants, since the programmable nature of computer based systems has a number of advantages over non-programmable systems. However, software is still vulnerable to common cause failure (CCF). Residual software faults represent a CCF concern, which threat the implemented achievements.

This study attempts to assess the effectiveness of so-called defensive strategies against software CCF with respect to reliability. Sensitivity analysis has been performed by re-quantifying the models upon changing the software failure probability. Importance measures then have been estimated in order to reveal the specific contribution of software CCF in the trip failure probability.

The results reveal the importance and effectiveness of signal and software diversity as applicable strategies to ameliorate inefficiencies due to software CCF in the reactor trip system (RTS). No significant change has been observed in the rate of RTS failure probability for the basic software CCF greater than 1×10^{-4}. However, the related Fussell–Vesley has been greater than 0.005, for the lower values.

The study concludes that consideration of risk associated with the software based systems is a multi-variant function which requires compromising among them in more precise and comprehensive studies.

1. Introduction

An anticipated operational occurrence followed by the failure of the RPS to trip the reactor eventuates in the Anticipated Transient Without Scram (ATWS), as defined in Appendix A.III of (NRC, 2007a), can lead to the unacceptable reactor coolant system pressures, fuel conditions, and/or containment conditions. The likelihood of core damage from the ATWS depends on three factors: (1) the initiating event frequency, (2) the reliability of the reactor protection system (RPS), and (3) the reliability of ATWS mitigation systems (NRC, 2007b).

The strong dependence of the ATWS risk on the RPS reliability and the uncertainty associated with the value of RPS unreliability were one of the major factors in the decision to adopt the ATWS rule (Raughely and Lanik, 2003).

A brief description on the issue of failure of trip function has been presented in Section 1.1. This section includes the software common cause failure description, its importance and in addition the software diversity.

Then, an introduction to importance and sensitivity analysis has been given is Section 1.2. This section has been aimed to discuss the application of importance measures, in addition to their definitions and related formula.

1.1. Introduction to trip on demand

The highly reliable RTS, as a part of RPS in nuclear power plant, is designed to sense the accident conditions and to initiate the rapid insertion of enough number of control rods.

Nowadays, software has increasingly being used to handle safety–critical system functions that were previously controlled

Abbreviations: AOO, anticipated operational occurrences; ATWS, Anticipated Transient Without Scram; CB, circuit breaker; CCF, common cause failure; CMF, common mode failure; CCCG, common cause component group; CRD, control rod; CRDM, control rod drive mechanism; D/P, diverse/primary; DRPS, diverse reactor protection system; DTM, digital trip module; FRT, final trip signal; FTA, fault tree analysis; FTS, final trip signal; FV, Fussell–Vesely; I&C, instrumentation and control; IST, in-service test; MGSET, motor generator set; NPP, nuclear power plant; PRA, probabilistic risk assessment; PRPS, primary reactor protection system; PTS, partial trip signal; PWR, pressurized water reactor; RAW, Risk Achievement Worth; RPS, reactor protection system; RT, reactor trip; RTB, reactor trip breaker; RTS, reactor trip signal; SSC, systems, structures and components; STI, surveillance test interval; TLU, trip logic unit; UPS, uninterruptable power source; USI, unresolved safety issue.

* Corresponding author. Address: School of Engineering, Shiraz University, Zand Avenue, Shiraz, Iran. Tel./fax: +98 711 647 3474.
E-mail addresses: nema@shirazu.ac.ir, mrnema@yahoo.com (M. Nematollahi).

0306-4549/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.anucene.2013.01.049

by humans or hardware in the past including the reactor protection system (RPS).

While the benefits of using computers in RTS are deniable, they are gained only at a price. In fact, the programmable nature of computer based systems with the discrete logic, have a number of advantages over non-digital and non-programmable systems, i.e. to facilitate the achievement of complex functions, provide improved monitoring of plant variables, improving the operator interfaces, improved testing, calibration, self-checking and fault diagnosis facilities. The use of multiplexed bus structures may lead to a reduced need for cabling. Software modifications require less physical disruption of equipment, which can be useful for maintenance (IAEA, 2000). In contrast, since software (other than the simplest programs) in its coded state or its compiled machine language state cannot be proven to be error free, residual software faults represent a primary CCF concern, which defeat the redundancy in the software based systems.

One hundred and forty one specific events have been reported, by Bickel (2008), for the accumulated operating experience, from 1984 to 2006 (1.27 billion hours), with the first generation of digitized RPS, on seven US nuclear power plants. Among them, 26 involved some types of common cause failure mechanism which temporarily degraded redundant portions of the overall trip function. Six of the common cause failure events were more severe and involved situations where incorrect addressable constant data sets were systematically loaded into all redundant computer channels due to personnel errors. One of these events involved a latent software design change error introduced during a software update, which would prevent proper operation, given an unlikely event involving failure of three out of four sensors of one type.

As a result, digital I&C systems receive particular emphasis in assessments of CCF susceptibility, resulting in application of techniques for avoiding or mitigating the potential for CCF vulnerabilities (Wood et al., 2010). About the I&C system being digitalized, three issues are encountered: (1) software common-cause failure, (2) the interaction failure between operator and digital instrumentation and control system interface, and (3) the non-detectability of software failure (Huang et al., 2008).

In fact, installation of the same software in redundant systems might defect the redundancy effect. The CCF probability of processor modules depends on the hardware failure probability of the processor module itself, the software failure probability, the diversity of processor modules, and the interaction effect between hardware and software.

Software diversity can be introduced to overcome the problem of coincident software failures in redundant parts of a computer based system. Although several methods have been suggested to introduce diversity into the development process, such as: diverse programming languages, diverse problem-solving algorithms, independent development teams and diverse tools, and their feasibility has been presented (HSE, 1998), there are still many different sources of potential coincident software failures and research in this field shows that statistical independence cannot always be assumed. In other words, there is currently no definitive guidance specifying how much diversity is sufficient to mitigate CCF vulnerabilities that may arise from the digital safety system designs (Dahli et al., 1990).

Therefore, despite uncertainties on the software failures, the sensitivity analysis research has been performed, to study the importance software CCF, based on the procedural steps described in methodology.

1.2. Introduction to importance and sensitivity analysis

The purpose of an importance evaluation is to identify the important basic events with regard to the occurrence of the undesired event. In other words, safety significance of systems, structures, components and human actions for preventive safety assurance activities are performed based on the values of risk importance measures (Wall et al., 2001). In this way, utilities hoped to focus their in-service testing and thereby reduce operational and maintenance costs while maintaining or improving safety, by using risk information to delineate between high and low risk significant components.

In many applications, only one risk importance measure could be sufficient, which is chosen based on the application, i.e. the FV importance measure as a measure of risk importance and RAW as a measure of safety-importance (Vaurio, 2001; Van der Borst and Schoonakker, 2001).

The Fussell–Vesely importance measure is expressed in relative terms. It indicates the risk associated with a given basic event E. That is, "how much this component or event is contributing to system failure".

$$I_{FV} = \frac{\text{sum of cutset contributions containing basic event}}{P_{Top}} = \frac{\Sigma_i \text{Cutset}_i(E)}{P_{Top}} = \frac{P_{Top} - P_{Top}(0)}{P_{Top}} \quad (1)$$

where P_{top} is the probability of occurrence of event i, in the base case and $P_{top}(0)$ is the probability of occurrence of event i, if its failure probability sets to zero (for example the component would be replaced with a perfect component).

FV is proportional to the unavailability of the component and represents the direct effect of the component unavailability on the unwanted event.

The Risk Achievement Worth measure is expressed as a ratio giving the factor by which the top event probability increases due to a component not being available, i.e. the event occurs with certainty, to assess which elements are the most crucial in maintaining the current level of reliability or availability.

$$\text{RAW} = \frac{P_{Top}(1)}{P_{Top}} \quad (2)$$

RAW is a weak function (almost independent) of the unavailability of the component. Therefore, RAW represents the defense of the rest of the installation against a failure of component, instead of the component itself. A large RAW reflects a strong defense in depth for the component in question (Schuller, 1997).

Importance measures and risk significance apply to events not to components. Regarding the criteria in Table 1, events with FV importance measure smaller than 0.005, are candidates for either no or very small in-service testing requirements. In contrast, components with FV $\geqslant$ 0.005 are currently risk significant and any degradation in their reliability would be significant. Such components rate an effective IST program. In addition, components, whose RAW importance are not smaller than 2, significantly increase the top event failure probability, when they are out of service, even though their contribution to the failure probability may be insignificant in long term (ASME, 2009).

2. Methodology

The methodology section presents the details of analysis to assess the software common cause failure and the effectiveness of mitigation strategies, following the stepwise procedural described below:

Table 1
ASME criteria for categorization of components (ASME, 2009).

Category	FV	RAW
High	>0.005	>2.0
Low	<0.005	<2.0