Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 28 (2014) 465 – 472

Procedia Computer Science

Conference on Systems Engineering Research (CSER 2014)

Eds.: Azad M. Madni, University of Southern California; Barry Boehm, University of Southern California; Michael Sievers, Jet Propulsion Laboratory; Marilee Wheaton, The Aerospace Corporation
Redondo Beach, CA, March 21-22, 2014

# Method for Generating a Diverse Set of Requirements for Safety-Critical Systems

Joseph Maurio[a]*, Christopher McClure[a]

[a]*Northrop Grumman Corporation, 7310 Sykesville Road, Sykesville, MD 21157, USA*

**Abstract**

Automatic digital safety-critical systems are often architected with redundant hardware in order to combat the effects of a single failure that could prevent the system from performing its safety function. Additionally, diverse hardware and software are typically employed to guard against any potential common-cause failures that would likewise cause an inability of the system to carry out its safety function. An all digital (processor or programmable logic-based) implementation usually requires the development of two digital systems by two separate software (and frequently hardware) teams which operate in parallel to provide the safety function. Strict rules are applied to the development process to ensure that the separate teams do not share information or influence each other's designs. Even though this technique provides a means to develop a diverse set of digital safety-critical equipment, the system design still begins with a single set of requirements. Therefore, it is conceivable that the two design teams may create solutions that contain identical design elements. Any flaws or vulnerabilities in the common elements would then be shared between the two designs making the system vulnerable to common-cause failures thus defeating the benefit of utilizing diverse design teams.

A method is proposed herein to address this limitation. This method entails the classification of the individual requirements of the source specification according to a detailed hierarchical taxonomy and the subsequent altering of the classified requirements. The taxonomy is structured so that the leaf-level classifiers are mutually exclusive or uncorrelated and the classified requirements are altered to be more stringent. The original and constrained requirements are allocated to two specifications documents in such a way that for certain requirements, the original version appears in the specification for one design team and the constrained version appears in the specification for the other. By using this process, sufficient requirements diversity results increasing the likelihood the two separate development teams will achieve a greater degree of design and implementation diversity than two teams using the same set of requirements. This increased product diversity should ultimately

* Corresponding author. Tel.: +1-410-552-2948.
*E-mail address:* joseph.maurio@ngc.com


Selection and peer-review under responsibility of the University of Southern California.
doi:10.1016/j.procs.2014.03.057

result in fewer latent common-cause faults residing in the two diverse systems. Furthermore, the degree of diversity achieved is expected to be greater when requirements diversity is employed, as compared to a traditional approach in which diversity is achieved by chance.



*Keywords:* requirements; taxonomy; diversity; safety-critical

## 1. Introduction

Diversity between redundant systems is required for many digital safety-critical applications found in industries where failures of these systems can lead to injury or loss of life. A prime example is the nuclear power industry where the Nuclear Regulatory Commission (NRC) has strict requirements and guidance on how digital safety-related systems are designed. The NRC's position is that "Digital instrumentation and control (I&C) systems can be vulnerable to common-cause failures caused by software errors, which could defeat the redundancy achieved by hardware architecture." [1]. Given this position, any digital (processor or programmable logic-based) safety-critical system within the nuclear industry must contain both redundant and diverse design elements. The typical approach to achieve the greatest diversity of systems for a particular safety-critical application in this industry is to utilize two or more separate design teams operating independently from each other. This independence is intended to minimize the possibility of the teams implementing identical hardware and software solutions. One weakness of this approach is that the separate design teams are given the same set of requirements. Arguably, this is unavoidable as the end product from each team must satisfy the same core needs and requirements of the customer for the equipment. It is likely that certain design elements will be identical between the two implementations since the two design teams work from the same set of requirements, leaving the system vulnerable to common-cause failures. This paper presents a method for overcoming this issue by defining a technique to generate two or more diverse sets of requirements from a single source set of requirements.

This work combines two separate techniques to generate a diverse set of requirements from a single set of requirements. The first is the classification of each requirement to a hierarchical taxonomy where the lower level classifiers are each uncorrelated with every other classifier in the taxonomy. The second is the constraining of each classified requirement so that it is unlikely that the altered requirement will be satisfied in the same way as the original equipment.

**Nomenclature**

| | |
|---|---|
| :: | subclassifier |
| \| | multiply-classified by |

## 2. Existing Methods

No prior literature could be found regarding systematic requirements diversification though requirements classification. Prior art exists regarding the classification of requirements in general[2, 3].

An automated approach for detecting and classifying non-functional requirements is presented by Cleland-Huang et al[4]. The disclosed technique is capable of detecting aspects or cross-cutting concerns from various project documents and measuring the accuracy with which the classification has been made. A flat set of classifiers is used rather than a hierarchical taxonomy as disclosed in this work.

Bitsch presents a formalism for expressing safety requirements using Computation Tree Logic (CTL)[5]. A "catalogue" of formal logic predicates is introduced that can be easily mapped to natural language safety requirements by software engineers who are not experts in formal specification languages, thereby "classifying" said requirements based on their temporal safety properties. This method is primarily aimed at proving safety properties of software-intensive systems but does not address diversity.