

A Quantitative Safety Assessment Methodology for Safety-Critical Programmable Electronic Systems Using Fault Injection

Michael A. Reynolds, Carl R. Elks, Nishant George, Meenakshi Sekhar,
Todd DeLong and Barry W. Johnson
University of Virginia

Copyright © 2009 SAE International

ABSTRACT

Given the increased use of programmable embedded electronic systems (PEES) in automotive applications and their vital importance, it is not only important for engineers to design PEES in such a way to meet or exceed safety requirements but also quantify how “safe” these systems are. At the University of Virginia’s Center for Safety-Critical Systems, we have developed a safety quantification methodology for embedded real time safety-related systems. The goal of the safety quantification methodology is to provide a generic but rigorous and systematic way of characterizing the dependability behavior of embedded systems that is applicable to a broad range of applications from automotive to nuclear. This paper presents a quantitative safety assessment methodology for safety-critical embedded systems using fault injection (FI). This methodology has been developed, refined and applied to a number of commercial safety-grade systems in the railway, nuclear and avionics industries. Additionally, we present several novel techniques that we developed to overcome long-standing challenges associated with fault injection based safety assessment.

INTRODUCTION

Automotive safety related systems based on electro-mechanical, pneumatic technology are rapidly being replaced by digital or information-based systems that contain complex devices such as embedded microprocessors, Application Specific Integrated Circuits (ASIC’s), Systems on a Chip (SOC), and Field Programmable Logic Devices. Collectively these

systems are called Programmable Embedded Electronic Systems (PEES). It has been shown that these systems offer great advantages such as tremendous processing power, flexibility, and low power consumption which enable more opportunities or strategies to reduce emissions, improve fuel performance, and improve safety and functionality with modest design and development costs. As more functionality in vehicle electronic systems migrates to PEES, important safety design and assessment issues naturally arise. Because of this increased use, there are mounting concerns on the overall vehicle safety to integration complexity, process technology maturity, design flaws, design and development assurance methods, and development tools for these devices.

A brief survey of Instrumentation and Control (I&C) embedded computer systems for automotive x-by-wire designs show they make extensive use of advanced technology (i.e. equipment, technology, and design practices) [1]. These designs are significantly and functionally different from traditional electro-mechanical, analog and discrete designs, and almost always include the use of microprocessors, programmable logic devices, A/D and D/A conversion electronics, displays, power conditioning electronics, real-time data busses, and multiplexing. Furthermore, these new I&C systems employ different fault isolation and containment techniques to achieve sufficient independence and fault tolerance than their predecessors. Taken all together, the introduction or consideration of new I&C systems into automotive x-by-wire applications has raised concern regarding the possibility that the fielding of these I&C systems, may introduce unknown or

unanticipated failure modes that have been overlooked, or a design error in the software in the redundant channels of a safety system which could lead to a common mode failure of the safety system function.

In order to prevent such situations from arising, there is a great need for these systems to be dependable across a wide spectrum of perceived threats, faults, failures; that is, these systems must be able to detect faults and initiate appropriate mitigation actions upon the detection of these faults to ensure overall safety of the vehicle. In recent years, significant research and effort has gone into design methodologies of safety critical systems, assessment methods, and upgrading regulatory and standards guidelines (e.g. IEC 61508, MISRA, RTCA-DO-254, etc) to ensure that I&C systems can be sufficiently designed and assessed to the high safety requirement levels required for SIL-level 3 and 4 systems. We argue from an engineering point of view, any assessment process or methodology of safety-critical systems should address the following questions:

- What causes a system to be less than 100% trustworthy¹?
- How should trustworthiness that is less than 100% be quantified?
- How should a required trustworthiness be achieved and assessed?
- What is justifiable evidence that a system meets its dependability requirement?

The safety quantification methodology developed at the University of Virginia, specifically addresses these questions. The goal of the safety quantification methodology is to provide a generic, systematic way of characterizing the dependability behavior of embedded systems (e.g. I&C systems) in the presence of anomalous behavior, faults, and failures [2]. The key aspect of the methodology is that it fundamentally represents a practice that recognizes the agreement or relationship that exists between certification entities, assessors, and vendors to develop reliable and safe I&C systems. From previous research efforts with the rail, nuclear, and avionics industries, we recognize the key features of this agreement should state the safety metrics that are pertinent for a given domain, how those metrics are quantified with supporting data, the way the data for the metrics are obtained, and the domain in which the measures and metrics are valid and meaningful. The end goal (for all parties) is a methodology that provides practical ways to characterize I&C system dependability that will facilitate developers in improving their products, and to help certification entities make informed decisions about the installation and purchase of I&C systems in critical application areas. In the next sections, we discuss the safety quantification methodology, dependability concepts, fault injection, and

the application of safety and reliability modeling to real world I&C systems. We begin by first covering some preliminary concepts to set the foundation for describing the methodology

PRELIMINARIES AND BASIC CONCEPTS

In this section we briefly cover the basic concepts needed to understand the context of this work. These concepts include, dependable systems, definitions, safety metrics, and fault injection.

CONCEPTS OF DEPENDABLE SYSTEMS – A dependable real-time system has the ability to provide its intended, expected and agreed upon functions, behavior, and operations in a correct timely manner [3]. All systems (including safety critical I&C systems) eventually experience failures. A failure is the inability of a system to deliver expected and agreed upon service. Understanding the relationship between the cause of failures (faults), the manifestation of faults (errors), and the erroneous behavior of the system (failure) is essential for appreciating how safety and/or reliability is assessed with respect to system requirements. This is why we make the strong distinction between faults, errors, and failures. In the dependable systems community, the Laprie model of dependability is widely recognized as the standard for understanding the concepts of dependability [3]. Dependability may be defined as the trustworthiness of a system such that reliance can justifiably be placed on the service it delivers [3]. A service delivered by a system is its behavior, as its users perceive it. A user is another system (human or technical), which interacts with the former. For a more comprehensive introduction to dependability concepts, refer to [3,4]. Figure 1 shows the relationships between the factors in dependability.

THE ATTRIBUTES OF DEPENDABILITY – The attributes of dependability are the primary means by which we specify the quantitative and qualitative requirements of a system. We define some basic terms and concepts related to dependability attributes.

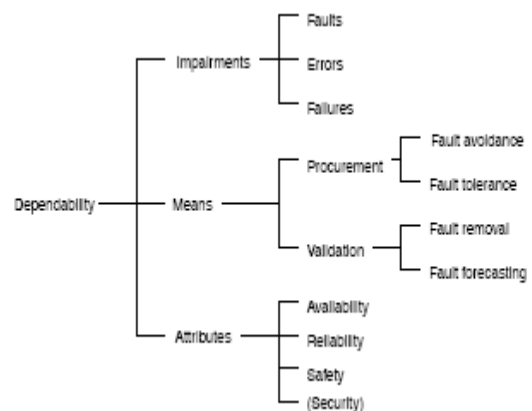


Figure 1 – Dependability Tree Linking the Various Attributes of Dependability [3].

¹ We assume that most real systems are not perfectly reliable or safe over their lifetime. Some measure of risk is always present, and the informed management of that risk is the ultimate goal of an assessment process.

Definition 1: Dependability, the property of a computer system such that reliance can justifiably be placed on the service it delivers [4], which is a qualitative system attribute that is quantified through the following terminologies.

Definition 2: Reliability, a conditional probability that the system will perform correctly throughout the interval $[t_0, t]$, given that the system was performing correctly at time t_0 , which is related to the continuity of service [4].

Definition 3: Availability, a probability that a system is operating correctly and is available to perform its functions at the instant time t , which is related to readiness for usage [4].

Definition 4: Safety, a probability that a system will either perform its functions correctly or will discontinue its functions in a manner that does not disrupt the operation of other system or compromise the safety of any people associated with the system [4], which is related to the non-occurrence of catastrophic consequences on the environment.

Definition 5: Fault coverage, C is the conditional probability given the existence of a fault that the system detects and recovers from that fault [4].

IMPAIRMENTS TO DEPENDABILITY – Faults, errors and failures affect the system's ability to deliver its dependability attributes (e.g. safety, reliability, performance). Hence, they are called the impairments to dependability. From our experience, there are important reasons to establish and evaluate the hierarchical, causal relationships between impairments, together with their origins and effects. First, it enables systematic quantitative and qualitative modeling and analysis of the reliability and safety of the system. Second, it serves as the basis for specification and requirements of an assessment of a dependability metric. We begin by defining the meaning of these expressions more precisely.

Definition 6: A **fault** is a physical defect, imperfection, or flaw that occurs within some hardware or software component⁴. A fault is the cause for an error.

Definition 7: An **error** is the manifestation of a fault. Specifically, an error is a deviation from accuracy or correctness [4]. Alternatively, an error is a design flaw or deviation from a desired or intended state.

Definition 8: A **failure** is the nonperformance or inability of the system or component to perform its intended function for a specified time under specified environmental conditions [4].

Implicit in the definitions of the above terms is a cause-effect relationship. The well-known 3-universe model depicted in Figure 2 shows the relationship between faults, errors, and failures.

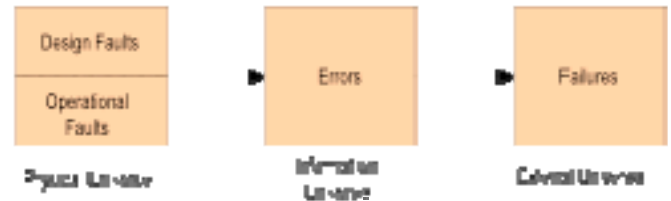


Figure 2 – Cause-Effect Relationship Among Faults, Errors, and Failures using the 3-Universe Model [4].

Essentially,

Figure 2 expresses a relationship that shows that failures are caused by errors, which are caused by faults. Associated with each term is a domain of effect, for example, faults are associated with the physical universe. Two categories of faults are possible in the physical universe: operational faults and design faults. Operational Faults include faults associated with semiconductor devices, mechanical elements, power supplies, and other physical entities that make up a system. These types of faults usually include fault types such as signal crosstalk in IC's, transistor failures in IC's, wear-out, oxidation, etc.

On the other hand, design faults include hardware and software flaws that are usually due to the inability to anticipate or fully consider certain interactions in hardware and software during system specification, design and implementation [4]. Design flaws or faults are activated when specific input stimulus and computer states are present, e.g. a unique execution path is taken by the computer-based machine element. Design flaws are not randomly occurring actions per se; they are deterministic events since they occur every time the same input and state conditions happen.

Errors are associated with the information universe, which contains units of information [4]. More precisely, information in a computer is characterized by symbols, and the interpretation and manipulation of those symbols. Errors can corrupt symbols rendering them into different symbols, non-symbols or reconstitute the interpretation of symbols. Errors in the information universe are usually manifested as bit flips in the data and/or instruction symbols. Finally,

Figure 2 shows that failures are associated with the external universe, which is where the user of the system eventually sees the effects of faults and errors.

For safety critical systems, failures associated with computer-based I&C systems that lead to hazardous conditions must be extremely improbable. Computer-based I&C systems without any additional protection against failures are most likely to fail in ways that are unacceptable to the overall safety of the system. Thus, a defensive strategy must be employed in computer-based elements to do one or two things to prevent the

occurrence of erroneous behavior associated with the I&C system:

- A **fail-safe action**: Upon detection of an error (or fault) that could lead to failure, force (by some defensive measure) the computer-based element to discontinue its operation in a manner that does not compromise the safety requirements of the overall system.
- A **fail-operational action**: Upon detection of error (or fault) that could lead to failure, overcome (by some corrective action) the erroneous condition such that the operation of the computer-based element is able to continue according to its specified safety and functional requirements.

Fail-safe and fail-operational actions are primary tenets for safety critical systems. These actions are not innate in most computer-based I&C systems; the computer must be designed to behave as a fail-safe or fail-operational system. Traditionally, fault detection and/or fault tolerance mechanisms are added to the fault intolerant design to detect and possibly correct the effects of faults during the operation of the machine element. Testing and validating the effectiveness of fault tolerance and fail-safe mechanisms within computer-based I&C systems is a fundamental part of the safety assessment process.

OVERVIEW OF FAULT INJECTION

Fault Injection is defined in [5] as the dependability validation technique that is based on the realization of controlled experiments where the observation of the system behavior in the presence of faults is explicitly induced by the deliberate introduction (injection) of faults into the system. Artificial faults are injected into the system and the resulting behavior is observed. This technique can speed up the occurrence and the propagation of faults into the system for observing the effects of faults on the system performance and behavior. Fault injection is a means to produce, measure, and estimate critical parameters in the analytical safety and reliability models. These parameters include, fault coverage, fault latency, mean time to reconfiguration, mean time to degrade, and error detection times.

Fault Injection can be performed on either simulations and models or working prototypes or systems in the field. It is better used to test the resilience of a fault tolerant system against known faults, and thereby measure the effectiveness of the fault tolerant measures. Fault injection techniques can be used in both electronic hardware systems and software systems to measure the fault tolerance of the system. For hardware, faults can be injected into simulations of the system, as well as into implementation, both on a pin or external level and, recently, on an internal level for some chips. For software, faults can be injected into simulations of software systems, such as distributed systems, or into

running software systems, at levels from the CPU registers to memory to disk to networks.

FAULT SPACE – In the assessment of digital or computer-based systems it is important to understand the context of a fault space. This is necessary because it is anticipated that I&C systems will experience failures, caused by a variety of faults. I&C digital systems are expected to detect these faults and continue operation for a specified number of fault occurrences over a time interval. There is the notion of fault tolerance. Fault injection or fault simulation are the backbone of testing the compliance and robustness of fault tolerance mechanisms in dependable I&C systems.

We use a fault space, $\mathcal{F}$, to describe faults [6]. A fault space, which in most cases is discrete and finite, is a simplified description of the faults that can occur in a digital I&C system. $\mathcal{F}$ (Figure 3) is often defined as a multi-dimensional space whose dimensions may include the time of occurrence and the duration of a fault (when), the type of the value or the form of a fault (how), and the location of a fault within the system (where). A value that could be something as simple as an integer or something much more complicated (such as state dependent fault) can be assigned to a fault. In general, it is very difficult, if not impossible, to completely prove the sufficiency of the modeling approach used to generate $\mathcal{F}$. Therefore, it is more traditional to assume that the fault space is sufficient, justifying this assumption to the greatest extent possible with the experiment data, the historic data, or the results published in literatures [4].

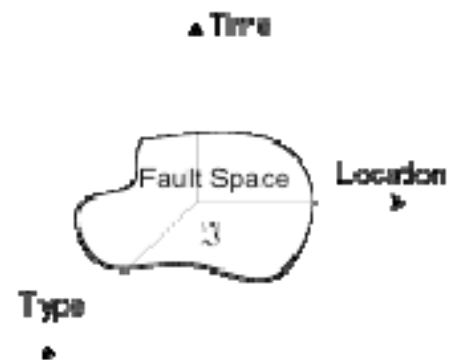


Figure 3 – A Fault Space

FAULT SPACE PARTITION – A fault space can be partitioned to two sub-spaces: Sub-space 1, which contains the safe faults, and Sub-space 2, which contains the unsafe faults (Figure 4). Safe faults cause a system to invoke a fail-safe action (e.g. a pump is forced to open position) that is also a non-operational state that will not cause a mishap [7].



Figure 4 – Fault Space Safe faults/Unsafe faults Partition

The consequences caused by safe faults can usually be recovered. Based on Carter's coverage definition (Definition 5), we use the safe fault and unsafe fault concept to distinguish whether a fault is recoverable. Assume a fault occurs at a time, t . We are able to derive:

$$\begin{aligned}
 C &= \frac{\text{Persistence of a safe fault at } t - \text{Persistence of a fault at } t}{\text{Persistence of a safe fault at } t + \text{Persistence of a fault at } t} \\
 &= \frac{\text{Persistence of a safe fault at } t}{\text{Persistence of a fault at } t}
 \end{aligned}
 \quad (1)$$

OVERVIEW OF THE METHODOLOGY

A key feature of the safety quantification methodology is the synergistic relationship between fault injection processes and analytical safety and reliability models of the target I&C system. Safety and reliability models allow for formal, repeatable and credible assessment of systems based upon measurable and observable parameters of the system under evaluation. The essence of the safety and reliability model is to allow quantification (e.g. calculation) of safety based upon known observable and measurable parameters of the system under assessment. The quantitative model based paradigm can best be described as a number of actions or steps.

1. Develop a mathematical reliability and safety model of the system
2. Measure or estimate the parameters of the model
3. Compute system reliability and safety based upon the model and the specified parameters
4. Determine the sensitivity of the model to parameter variations and model uncertainties.
5. Collect empirical data to support the predications of the model.

When you can measure what you are speaking about, and express it in numbers, you know something about it.

Fault injection stresses the fault handling and management aspects of the system under assessment to ultimately collect crucial data via fault injection campaigns. In most fault tolerant systems, the FDIR (Fault Detection, Isolation, Recovery) software accounts for 40%-50% of the executable system software code [8,9]. This code is rarely exercised in the real world

because faults are infrequent occurrences. This FDIR code is vital for system compliance to dependability, and can only be effectively tested and validated by realistic fault injection campaigns. The general approach is to inject randomly selected faults into the system to determine if the fault processing capabilities detect, tolerate and mitigate the faults. This data collected during fault injection is used to statistically and accurately estimate the crucial parameters of the analytically safety model. These parameters typically include fault class coverage, recovery time of a fault, fault latency, error detection coverage, and error detection time. Once these parameters have been accurately estimated, then they are instantiated into the safety models to accurately calculate the safety of the system. The safety and reliability quantification approach described in this proposal builds upon the University of Virginia's extensive research and applied experience in this area. A detailed report can be found in [10]. The methodology is outlined in Figure 5 and consists of the following steps:

Step 1 – An analytical safety model is developed from the system architecture and inter-component dependencies to derive an expression for the safety. As an example, we can use Mean Time To Unsafe Failure (MTTUF) [11]; however, other safety expressions such as probability of coincident failure, probability of uncontrolled event, or probability of missed real-time deadline can be used. In fact, the methodology may use several safety models or expressions to determine the compliance of a system with respect to its dependability requirements. These models are typically represented as Dynamic Fault Trees, Markov models, or combinatorial models. The safety expression is a function of Critical Model Parameters such as the fault coverage values, recovery rates, and critical failure rates for the various components of the system. For the purposes of a safety assessment, the analytical model is used to model, at a high level, the faulty behavior of the system under analysis. The model uses various critical parameters such as failure rates, fault detection latencies, and fault coverage to describe the faulty behavior of the system under analysis. Of all the parameters that typically appear in an analytical model used as part of a safety assessment, the fault coverage is by far the most important and difficult to estimate. One of the main purposes of fault injection is to accurately estimate this parameter.

Step 2 – A statistical model is then developed based on those found in published literature and is used to estimate the Critical Model Parameters that are required by the analytical safety model. This statistical model is used to calculate the number of fault injection experiments required to meet the numerical safety target for a given confidence level. In essence, the statistical model determines how many fault injections or specific fault types are required to accurately estimate the overall system safety. See reference [12] for more details.