Software Risk Assessment: Fuzzy Logic Approach to Risk Estimation (FLARE)

Willie Fitzpatrick, PhD; US Army AMRDEC Software Engineering Directorate; Redstone Arsenal, AL, USA

David Skipper, PhD; SAIC; Huntsville, AL, USA

Josh McNeil; US Army AMRDEC Software Engineering Directorate; Redstone Arsenal, AL, USA

J.P. Rogers; APT Research, Inc.; Huntsville, AL, USA

Keywords: Possibility Theory, Fuzzy Logic, System Safety, Airworthiness, Software Safety, Risk Assessment, Software Risk Assessment

Abstract

Industry standard methods for hardware and operations risk assessments include hazard severity and hazard likelihood in risk predictions. Software "hazard" assessments include the same hazard severity employed in hardware and operations risk assessments, but use software control authority in lieu of failure likelihood (ref. 1) to determine a software safety assurance rather than a "risk". Since risk is estimated by the combination of event severity and likelihood, hazard assessments for software do not result in "risk". In the absence of a software risk, it is difficult to characterize the system level risk as a composite of hardware, operations, and software risk. To correct this deficiency, a qualitative estimate for the likelihood of software safety failures is needed. Attempts to determine the quantitative "failure probability" of software have not resulted in wide acceptance or consistent application. This paper presents an estimation formalism based on a fuzzy logic approach to software risk estimation. This approach utilizes current accepted software safety processes so that a fuzzy estimate for software safety failure likelihood can be combined with severity category to estimate software contribution to system level risk.

Overview

Risk is the product of hazard severity and event likelihood (ref. 2). Industry standard methods have been developed for hazard risk assessment. These methods are used for hazard risk assessments of hardware failures and operations errors. Standard hazard assessments include both the hazard severity and the event likelihood risk components. However, estimating software "risk" is not a standard activity in software hazard assessments (ref. 3). Predicting system level risk in terms of the composite of hardware, operations, and software risks is a desirable, but difficult objective, given the absence of a true software risk assessment (see Figure 1). This paper proposes a method to address the software risk assessment deficiency in the system level risk assessment. A fuzzy logic based approach is employed to develop a qualitative likelihood of software safety failures. This qualitative likelihood is then combined with hazard severity to produce a fuzzy software risk level estimate that is qualitatively similar to the hardware and the operations risks levels (*e.g.* High, Medium, Low) (ref. 4). The hazard severity risk component is assumed to be developed from analysis performed prior to initiating the FLARE process. FLARE is then employed to estimate the missing software risk component - the likelihood of a software safety failure event (see Figure 2). The FLARE process is currently being assessed using actual system and safety data.

Assessing hazard severity in linguistic terms (*e.g.* catastrophic, critical, etc.) is a straightforward activity (ref. 4). However, estimating the likelihood of a software safety failure in a Safety Significant Function (SSF) is not a trivial process (ref. 3). Software safety assessments are historically not probabilistic in nature. They are based on individual decisions analysts make. The assessment is evidence/artifact driven and it reflects the analyst's confidence or belief in the "goodness" of the software's safety characteristics (ref. 5) relating to software failures. The analyst's confidence or belief is then the basis for developing risk likelihood. The Software System Safety discipline has adopted a safety assessment process for analysts that use both software hazard analysis objectives and software development objectives that are designed to reduce the likelihood of software safety failures (refs. 6, 7). These are the analyst's primary evidence/artifacts and they are used to increase/decrease the analyst's belief that the

Thank you for your interest in our papers!

For the rest of the paper, please email aptinfo@apt-research.com