Scalable Software Evaluation Methodology and Tools

A JFTP Proposal for FY12 Under FATG IV

Jeffrey M. Fornoff
US Army RDECOM-ARDEC
Picatinny Arsenal, NJ 07806-5000
(973) 724-3014

Purpose

- Fuzing designs are utilizing software in part or in whole in the implementation of one or more safety features in networked environments
- Safety critical software needs to be analyzed and tested to assure proper design and function
- No documentation exists to help identify all issues related to networked safety critical software in munitions.
- AOP-52 provides guidance in the requirements of design and implementation of safety critical software throughout the entire life-cycle, but not enough network guidance exists

Background

- Implementation of Safety Features has evolved significantly
  - Mechanical (springs, setback weights, rotors, sheer pins)
  - Electronic (analog and/or simple digital circuits)
  - Software (microprocessor and/or programmable logic devices)
- The use of software has allowed designers the ability to design fuze safety features that are influenced partially or completely outside the fuze itself. The resulting "fuze chain" blurs the line between the fuze and the rest of the munition
- Recent standards and guidelines were developed to address software safety in fuzes, fuzing systems, and munitions - AOP-52 was developed under NATO AC326 Sub-group III with members of the US (DoD), UK, Germany, and France

Case History

- Distributed fuze functions first encountered by the AFSRB with Spider hand-emplaced anti-personnel landmine
  - Arming completely controlled by a Remote Controlled Unit (RCU) (laptop PC) communicating with the Munitions Control Unit (MCU)
  - On-Off-On requirements made the software on the RCU safety critical (in addition to the software on the MCU)
  - Due to lack of suitable documentation at the time, software safety guidance was developed concurrent with the system design
- Lessons learned helped produce a better safety design with Scorpion
  - FESWG Guidelines on the use of Programmable Logic Devices
  - AFSRB Guidelines Appendix C
  - AOP-52

Current Challenges

- Networked (remote controlled) munitions are becoming the "norm" given technology available today
- System of Systems allow remote arming/firing of weapons half-way around the world
- How does safety get designed into these advanced systems?
- What design parameters need to be addressed?
- What metrics can be identified to verify and test safety critical features?
- Bring to bear any existing software analysis techniques and/or tools – develop them if they don't exist

Proposed Work

- Identify safety critical parameters in a networked environment
    - Protocols (important for guaranteed communication)
    - Latency (important for real-time operations)
    - Quality of Service (QoS) (important for reliability & safety)
    - Data Integrity within networked communications and software
    - Information Assurance (IA) as it relates to safety
- Identify System of System integration issues affecting safety
- Create/augment guidance documents (i.e. update AOP-52)

Proposed Work (continued)

- Perform research consulting with academia to identify safety critical parameters that need to be addressed in networked system designs
- Create a straw-man documentation that can be used to update AOP-52
  - Create a working group of SMEs from all the services
  - Work with NATO AC326 Sub-group III to update AOP-52
- Identify analysis techniques and/or tools that can be utilized to verify safety critical functions are properly designed and implemented

Summary

- Research and identify networking parameters that affect software safety with the help of academia
- Update AOP-52 with respect to networking issues and provide additional guidance topics on System of Systems
- Identify software metrics affecting safety in network systems
- Identify and/or develop software analysis tools that can be used to test and verify safety critical metrics in network systems
- Identify scaling techniques and safety critical parameters when interfacing software in System of Systems

Questions?

Backup Slides

AOP-52 Contents

- Executive Overview
- Introduction to the AOP
- Generic Guidelines and Requirements
- Software Safety Engineering
- COTS and NDI Software
- Testing and Assessment Guidelines
- References
    - Software Development Models
    - Lessons Learned
    - Process Charts

Executive Overview

- Explains the need and importance of software safety elements in requirements and design of the program to managers and executives
- Software safety begins with program requirements and continues throughout the life-cycle of the program
- Even though software is only a part of the overall safety of a program, there are unique issues with software safety that need to be addressed

Introduction to the AOP

- Explains the major purpose for the AOP
- Presents the scope of the subject matter incorporated in the document
- How the AOP is organized for maximum benefit of use
- Also includes:
    - Historical background
    - Problem identification
    - Introduction to the “Systems” approach

Generic Guidelines and Requirements

- Introduction
- Determination of Safety-related Computing System Function
- Design and Development Process Requirements and Guidelines
- Software Design Verification and Validation
- System Design Requirements and Guidelines
- Computing System Environment Requirements and Guidelines
- Self-check Design Requirements and Guidelines
- Safety-related Computing System Functions Protection Requirements and Guidelines
- Interface Design Requirements
- Human Interface
- Critical Timing and Interrupt Functions
- Software Design and Development Requirements and Guidelines
- Software Maintenance Requirements and Guidelines
- Software Analysis and Testing

Software Safety Engineering

- Introduction
- Software Safety Planning Management
- Software Safety Task Implementation
- Software Safety Testing and Risk Assessment
- Safety Assessment Report/Safety Case

COTS and NDI Software

- Introduction
- Related Issues
- Applications of Non-Developmental Items
- Reducing Risks
- Testing
- Summary