Applying System Safety Axioms in the Evaluation of Complex Open Architectures

Mike Allocco, ISS Fellow

Introduction

In applying system safety axioms in the evaluation and design of complex open architectures system risk can be identified and controlled to an acceptable level.

The purpose of this paper is to provide an example of how classic system safety methods can be applied within complex distributed open architecture designs. By following system safety concepts industry and government-wide safety requirements can be established to enable the use of open designs.

Background

Computer architecture is the conceptual design and fundamental operational structure of a computer system, which includes hardware, firmware, and software elements. (From a system safety perspective the human and environment is also to be integrated into the evaluation and design). The architecture design process addresses the art and science of selecting and interconnecting hardware components to create computers that meet functional and performance objectives, (and system safety objectives). An open architecture includes specifications that are available to the general public. These are approved standards as well as privately designed architecture whose specifications are made public by the designers. The advantage of open architecture is that anyone can design integrating products for it. By making the architecture public, a manufacturer allows others to duplicate its product.

System Safety Challenges

An open architecture (OA) presents challenges from a system safety perspective in that there may be inherent system risks because of the use of commercial-off-the-shelf, government-off-the-shelf, and complex legacy systems that may not have met safety requirements; or because intended use will vary depending on newly designed add-on systems. Regardless of these challenges it is quite possible to apply system safety axioms to assure that the system risks have been eliminated or controlled to an acceptable level. (System risks are emphasized in that system-level safety requirements are needed to address not only software and firmware, but hardware, the environment, and human factor interactions.)

Example OA Problem and System Safety Process
Consider that a safety critical system is to be designed and the system is to be integrated within an existing open architecture. There are a number of steps to be taken from a “system” safety perspective, consider the following:

1 - The integrated system must be described in order to conduct integrated hazard analysis. The system is to be defined in various ways: functionally, architecturally, and operationally. Define all the outputs and inputs that communicate between the OA and new safety critical system. Decompose the interface and new system in various forms and understand sequences, event progressions, and threads. Develop models such as state diagrams, network flows, digraphs. Understand the intended use of the integrated system and also consider potential misuse of the system. Address software, firmware, and hardware, human and environmental interactions.

2 – Consider conducting integrated scenario-driven hazard analysis, evaluating hardware, software, firmware, human, and environment elements; in order to identify system risks.

Figure 1 below depicts the example safety-critical system that is to be integrated with (or within) an existing OA. There is a human computer subsystem and an automated subsystem. The human is to take contingency action based upon what is displayed on the safety critical laptop and the automated subsystem is also safety critical. There are a number of “system” risks that come to mind:

- Undetected hazardous misleading information presented to the human and the human takes inappropriate action hindering contingency action when needed. The situation results in a catastrophic outcome.
- The automated subsystem takes inappropriate action and inadvertently operates. The outcome is also catastrophic.
- There is delayed digital communication to the human during an emergency and contingency is also hindered.
- A safety-critical command input is needed to “Safe” the automated subsystem and the command signal is lost or delayed.
- The system is spoofed and access is gained by an unauthorized threat or intruder.
- The system is jammed or lockup occurs at a critical time.
- The human introduces a real-time hazard via human error any time during the life cycle of the system.
- Decision errors are made, which introduce latent hazards within the system: logic error, timing error, or sneak path for example.
- Product history is inadequate and latent hazards are not detected.
- Plug-in systems provide a use or function not foreseen and real-time hazards occur.
- Over complexity is introduced and system state status is not known.
Figure 1: Depiction of an example safety-critical system that is to be integrated with (or within) an existing OA.
3 – Apply many different hazard analyses methods in order to understand potential accidents (system risks). Consider that the potential accident is a form of adverse integration of many hazards: initiators, contributors, and primary hazards. These hazards may stem from: specification errors, judgment errors within the design, mistakes in development and coding, compiling errors, upsets, adverse energy effects, hardware and firmware failures, inappropriate human action, anomalies, malfunctions, logic errors, timing, scheduling, and sequencing errors\(^1\).

Depending on the system risk it may be appropriate to confine thinking to address so-called software or hardware hazards. When applying functional approaches understand how the functionally-related hazard may manifest by means of logic error, firmware failure, hardware failure, or human error. Appropriate detailed hazard analyses are also required to evaluate subsystem risks, via subsystem hazard analyses. Consider classic approaches such as:

- Software, Hardware Failure Modes and Effects Analysis
- Hazard Operability Analysis
- Thread Analysis
- Walkthrough Analysis

4 – Evaluate the human throughout the life cycle: the computer-human interface, command, control and communication tasks and procedures, contingency action, and maintenance actions.

5 – Apply system and current engineering concepts assuring that requirements are in concert with system safety: availability, logistics, quality, maintainability.

6 – Conduct threat and vulnerability analyses to assure that the security risks (that adversely effect system safety) have been eliminated or controlled to an acceptable level.

7 – Develop mitigation with the application of layering of hazard controls, which are to eliminate or control system risks; both engineering and administrative controls to act as barriers to hinder or abate adverse flow, (see the discussion below on the automated safety monitor).

8 – Consider past knowledge associated with similar systems: loss analysis, incidents and accidents. Include an understanding of the service history of commercial-off-the-shelf, government-off-the-shelf, and complex legacy systems.

9 – Monitor the system to assure continued hazard tracking and risk resolution.

10 – Evaluate any changes to the integrated system and conduct reevaluation.

Automated Safety Monitors (ASM) (or Safety Buffer)

Independent automated safety monitors (safety buffers) are a form of engineering control that can be used to safe-isolate the safety-critical system from the existing OA. As stated, there are challenges from a system safety view that faces the high-tech digital community today, which involve the integration of new technologies into the existing or newly designed open systems, which includes legacy systems, reusable software, Commercial off-the-Shelf (COTS) software, Government off-the-Shelf (GOTS) software, and Non-Developmental Items (NDI). There can be latent or real-time system risks as a result of inadequate system safety efforts.

End-to-end integrity must be verified and validated from a system safety viewpoint\(^2\). System safety efforts require extensive assurance activities associated with the system and software such as rigorous design and formal documentation\(^3\). Such efforts may be cost prohibited when considering a complex, distributed, loosely coupled open dynamic systems. An appropriately designed automated safety monitor may provide the means to eliminate or control these system risks to an acceptable level.

The primary objectives of the automated safety monitor are to:

- Monitor all safety-critical inputs and output within the interface between the OA and newly integrated safety-critical system.

- Provide for the optimization of status monitoring and alarm annunciation and communication including the filtering of false and unimportant alarms, the organization of alarms to reflect system safety priority and adverse relationships, and the provisioning of high-level functional alarms.

- Provide on-line fault diagnosis, hazard control and correction, including early detection of escalating initiators and the isolation of initiators from observed anomalous adverse sequences. Hazard control and correction is provided to the ASM operator to provide an assessment of the effects of disturbance and guidance on corrective actions that remove or minimize the effects of the problem; or automatically self correct the adverse malfunction.

---

\(^2\) Within hazard analyses, there may be open risk controls that require validation from an end-to-end system view. They relate to automation: hardware, firmware, and software safety assurance.

\(^3\) Radio Technical Commission for Aeronautics (RTCA) DO-278B, Guidelines for Communication, Surveillance, and Air Traffic Management (CNS/ATM) System Software Integrity Assurance, requires extensive processes and documentation.

Users are to be provided with real-time independent automated safety monitoring capability, which includes enhanced (system safety) situational awareness of the system status of the end-to-end distributed integrated system. At the system level, the objective is to make predictions about the propagation of a disturbance while the disturbance is in progress and still at the early stages. Such predictions are used to assess the severity of the disturbance and the risk associated with the current state of the distributed system. Proactive contingency action is to be taken to automatically and/or manually correct the situation before harm occurs.

There are many system safety-related requirements associated with the life cycle of the independent automated safety monitoring capability. The 10 step system safety process suggested above remains applicable to the safety buffer evaluation.

The Need for Industry and Government-Wide Safety Requirements for Open Architecture

Since the 1960’s there apparently has been a successful paradigm that has been implemented time and time again, considering the system safety practice. A new high risk advanced technology is initially contemplated, such as the use of nuclear energy, nuclear weapons, rockets, missiles, high performance aircraft, space craft, software computing, digital systems, and unmanned systems. There are the usual initial concerns and these concerns are openly discussed. Eventually individual system safety analyses are conducted at various projects and safety requirements are propagated. The next apparent step is the formation of an agency-wide system safety working group. The output of such activity has been the development of national and international safety standards and design guides for nuclear weapons, software engineering, space craft, specialized weapons, and unmanned systems.

An argument can be made that for OA individual system safety analyses have been conducted and are also underway, considering that safety requirements have been publicly discussed. It is recommended that the next logical step is to form an agency-wide system safety working group to vet and develop national and international safety standards associated with OA.

A Starting Point for OA Safety Requirements

An open architecture requires the integration of a common computing environment, common user functions, and unique user functions; consequently there are common physical, functional and operational safety requirements to enable system of system integration; consider including:

- Modular design to enable integration of common user functions, and unique user functions
- Modular and hierarchy segregation between high-level and low-level safety-critical functions

4 The suggested requirements for an automated safety monitor has been discussed and detailed within an upcoming book by the author: The Safety Analyses of Complex Systems: Considerations of Software, Firmware, Hardware, Human, and the Environment, to be published by Wiley Interscience.

5 Many of the controls that address OA design have been discussed within the following presentation: Naylor, W., Shank, B., McVae, L., Maintaining Safety in an Open Architecture Environment, System Engineering Advisory Group (SEAG) Symposium 2009, May 2009.
• On-going safety certification to enable system of system integration and safety assurance
• Common system safety validation and verification methods for software, firmware, hardware, environment and human-related hazard controls
• Consistent safety-related testing: regression tests, stress tests, decision tests, go/no-go tests, static and dynamic tests, integration tests, safety V&V tests, operational tests, functional tests
• Common standardized system safety analyses methods and techniques
• The acquiring data and information concerning: past history, service history, past use, intended use, accidents, and incidents
• Compatibility with interfacing operating systems
• Isolation and protection for safety-critical systems
• Assurance of continued integrity of safety-critical functions
• Assurance of communication to and from safety-critical systems: handshaking, security controls, conformation
• Limit and reasonableness checks
• Boundary checks, data expectation, format and value limitations
• Throughput, latency, and timing checks
• Health monitoring of system: hardware and firmware
• Self correction, self repair of safety –critical functions and systems
• Data flow sequence verification
• Failsafe designs in the event of failure, error, malfunction, anomaly
• Consistent data typing, naming conventions
• Standard interface protocol
• Architecturally-based testing, more testing based upon compatibility between computing environments, operating systems
• Memory protection
• Failure propagation prevention
• Real-time system state status indication
• Real-time feedback to operators
• Command and control protocols prior to initiation of hazardous function
• Use of interlocks, lockins, safety devices
• Protection against inadvertent operation
• Progressive quality engineering
• Configuration and logistical control
• Verification of human interface
• Physical protection from environment
• Multi-level hazard control
• Consistent documentation throughout system life cycle
• Behavioral-based safety controls throughout system life cycle
• Error tolerance controls throughout system life cycle
Summary

System safety work remains laborious since the devil is in the detail. Consider applying the recommendations discussed to assure that system risks have been eliminated or controlled. In applying classic system safety axioms in the evaluation and design of complex open architectures system risk can be identified and controlled to an acceptable level.

There is a need for detailed efforts to enable the identification of all safety-critical inputs and output to and from a complex open system. Automated safety monitors or safety buffers are high-level engineering controls to safe-isolate integrated safety-critical systems form loosely coupled dynamic distributed open systems. Designs should assure real-time hazard control validation and verification to prevent or abate adverse progression between safety-critical systems that are integrated with open systems.

Biography

Mike Allocco, PE, CSP, has been employed in Safety Management, System Safety, and Safety Engineering since 1976. He has conducted hazard analysis and risk assessments of nuclear and conventional weapon systems, the space station, various aircrafts, aircraft ground systems, medical devices, railroad systems, tunnel boring machines, complex processes, and facilities. Mike is coauthor (with Dev Raheja) of Assurance Technologies Principles and Practices – A Product, Process, and System Safety Perspective, Second Edition. He has conducted system safety engineering on diverse complex systems for the general industry, DOT, DOD, DOE, and NASA. Mr. Allocco is a ISS Fellow and was also a former Executive Vice-President of the System Safety Society; he is currently employed by the FAA.