RISK MANAGEMENT FOR CONTROL SYSTEM SOFTWARE

IMCA Annual Seminar
London, 2014-11-19

Øyvind Smogeli, COO
Bjørnar Vik, Principal Engineer
ISM - THE INTERNATIONAL SAFETY MANAGEMENT CODE

How does this relate to control system software?

- Ship owners should identify *equipment and technical systems* the sudden operational *failure of which may result in hazardous situations.*

  Failure of control system SW may most certainly lead to hazardous situations.

- The *safety management system* should provide for specific measures aimed at *promoting the reliability* of such equipment or systems.

  The safety management system cannot exclude software…

- These measures should include the *regular testing of stand-by* arrangements and equipment or *technical systems* that are not in continuous use.

  A lot of control system functionality – especially barriers and other failure-mitigating functions – lie dormant during normal operation.

The Safety Management System must include handling of control system software, including risk assessments and mitigations.
SOFTWARE IS FRAGILE: ARIANE 5 – JUNE 4\textsuperscript{TH} 1996

```plaintext
L_M_BV_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BV) * G_M_INFO_DERIVE(T_ALG.E_BV));
if L_M_BV_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsi
if L_M_BV_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BV_32));
end if;
end if;
L_M_BH_32 := TBD.T_ENTIER_32S ((1.0/C_M_LSB_BH) * G_M_INFO_DERIVE(T_ALG.E_BH));
if L_M_BH_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BH) := 16#7FFF#;
elsi
if L_M_BH_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BH) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BH_32));
end if;
```
-Eh, can you make it 3 feet wider?
Class Bridge

private double width = 60;
private double length = 4500;

public void tiltTowers45degrees();

- Sure, no problem. Change the length also?
WHY DOES SOFTWARE FAIL?

THE CURSE OF FLEXIBILITY…

- Software has no physical constraints that limits the flexibility and complexity of what we build.
- With software, the limits of what is possible to accomplish are different than the limits of what can be accomplished successfully and safely.
- We can construct complex software that goes beyond human intellectual limits, making it hard to understand how it will behave under all conditions.
- Software deliveries are unique and tailor-made (configured) for the vessel.
- A lot of SW functions and configuration is written under a tight time schedule and finished late in the projects.
- Control systems are complex with many signals and interfaces.
- Integration of systems from different vendors is challenging and error-prone.
- Frequent updates and patches introduce new problems.
- Different people modifying or repairing the system.
- Modifications and repairs tend to be made locally and cheaply.
- Deficiencies in documenting modifications.

2014 © Marine Cybernetics | www.marinecyb.com
SOFTWARE IS A CHALLENGE

- Power management system
- Crane control system
- Drill floor control system
- Well control systems
- BOP control system
- Thruster control system
- Emergency shutdown system
- Integrated automation system

High configurability
- OS, Firmware, Applications, Network, ++

Many vendors

10 000’s of functions

10 000’s of I/O

1000’s of functions

Many vendors

2014 © Marine Cybernetics | www.marinecyb.com
RISK MANAGEMENT PROCESS

- Establishing the context
- Risk identification
- Risk analysis
- Risk evaluation
- Risk treatment
- Communication and consulting
- Monitoring and review

- (ISO/IEC standard 31000:2009)
BOW-TIE

Preventive barriers

Potential incident

Top event

Stopped by preventive barrier

Mitigating barriers

Escalating incident

Stopped by mitigating barrier

2014 © Marine Cybernetics | www.marinecyb.com
SW BARRIER AS ROOT CAUSE THREAT

Preventive barriers

Preventive SW barrier

Hidden SW bug initiating a potential incident

Top event

Escalating incident

Mitigating barriers

Defective barriers

2014 © Marine Cybernetics | www.marinecyb.com
TRADITIONAL RISK ASSESSMENT TECHNIQUES

- **HAZOP- Hazard and Operability study** investigates how the system or plant deviate from the *design intent* and create risk for personnel and equipment. HAZOP is both a top-down (inductive) and bottom-up (deductive) technique.

- **FMEA – Failure Mode and Effects Analysis** is a bottom up (deductive) technique which looks at component failures and their impact on the system. Qualitative, identifies design areas needing improvement.

- FMEA can be extended with a criticality analysis (FMECA)
  - Usually qualitative (ranking according to some scale, e.g. 1-10 or low/medium/high)
  - Can be quantitative (failures/hour or failures on demand)

The DP FMEA, which is required for DP 2 and 3 vessels, has elements from both HAZOP and FMEA.
QUANTITATIVE RISK ANALYSIS (QRA)

- For high-risk effects, a quantitative risk analysis (QRA) is sometimes used in preference to a FMECA.
- QRA needs good historical or test data using industry databases or other sources.
- Important to consider the quality and relevance of data used for this purpose.
- Event Trees and Fault Trees can be used to break down the problem and calculate quantitative risk estimates, but the probability of failure must still be determined for each basic event or initiator.
SAFETY INTEGRITY LEVEL (SIL) FOR SOFTWARE

- SIL is a recognized method for qualification of safety systems. It is used for offshore drilling vessels and in other industries where safety is paramount, such as railway, automotive, and nuclear.
- SIL requirements based on IEC 61508 are quantitative for hardware while requirements for software are qualitative (based on development and test processes).
- It is important that SIL numbers are used with care.
- Functional and black-box testing are required for all SIL levels.
- Modification of software requires:
  - Impact analysis
  - Re-verification of changed software module
  - Software configuration management
  - Data recording and analysis
- Some nations allow equivalents to SIL through class (e.g. Norway)
CURRENT RISK MANAGEMENT IS HARDWARE BIASED

- There are well established processes for *independent* risk assessment of structures, noise & vibration, and electro-mechanical hardware systems.
- Risk assessment for safety critical systems is firmly rooted in the probabilistic analysis of failure events, but probabilistic analysis of software is controversial.
- DP FMEA does not cover software, and most software faults cannot be found using the traditional testing methods used onboard.

- How is control system software included in risk assessments?
  - Zero probability of failure (ignored)?
  - Included with very conservative numbers (high probability of failure)?
  - With limited failure modes, such as complete loss of function?
HARDWARE AND SOFTWARE RELIABILITY

- **Hardware reliability**
  - Failure rate drops during burn-in and stays constant throughout the lifetime of the component.

- **Software reliability**
  - Software does not wear out
  - Software failure rates can increase due to upgrades

The source of failures in software is design faults, while the principal source of hardware has generally been physical deterioration.
WHY CANNOT HW RELIABILITY QUANTIFICATION METHODS BE USED FOR SW?

- HW reliability can be tested by running the same test on several copies over and over until they fail.
- A number of theories can be applied to describe hardware failures (e.g. the physics of the failure process).
- Hardware is replaced when not working
- Redundant hardware can be assumed independent

- Software reliability needs to be tested using an operational profile.
- None of the theories for hardware failures can be used for software failures.
- Software is updated when not working
- Software is usually not redundant

- The usability of reliability models for software is debated due to the number and nature of the assumptions needed.
- A major issue with quantification is the focus on producing numbers. The assumptions and limitations behind these numbers, which may be questionable to begin with, can easily be lost.
DP2 VESSEL REDUNDANCY CONCEPT – HW VS SW

Standby controller:
- SW failure
- Same SW as main controller

Main controller:
- SW failure
- Same SW as backup controller change to backup controller

m A and B over
HOUSTON, WE HAVE A PROBLEM

- Risk management for control system software is a challenge
- How can we handle it?
- How to really comply with ISM?

- Improve risk assessment methods by SW specific techniques
  - SW FMEA, SW HAZOP (qualitative techniques)

- Improve software engineering processes
  - IEEE 12207, CMMI
  - ISDS and ISQM

- Improve testing and verification
  - IEEE 1012 and IEEE 29119
  - For control systems, Hardware-In-the-Loop testing is the main tool

- Introduce life-cycle software change management

- A combined approach will give the optimal solution:

  Software engineering, testing and verification, risk assessment and change management coordinated into a common software and system integrity service
WHAT IS HIL TESTING?
CONTROL SYSTEM IN NORMAL OPERATION - CONTROLLING REAL VESSEL
WHAT IS HIL TESTING?
CONTROL SYSTEM IN NORMAL OPERATION – CONTROLLING SIMULATED VESSEL
DP FMEA EXTENDED WITH THE SOFTWARE PART

Traditional DP FMEA

- DP FMEA desktop study
- FMEA trials
- Annual and 5 year FMEA trials

Extend with software FMEA and HIL testing

- Software FMEA
- HIL Software testing
- HIL ViO

Construction → Installation and commissioning → Operation

FMEA analysis → FMEA verification → Change management
ISDS AND HIL COMBINED

Test artifacts:
- Software Requirements Specification
- Software Design Specification
- Code

Test artifacts:
- Running code
- Hardware implementation
- Human-Machine Interfaces
- Integrated systems

HIL strength: Product

Process focus

Static test methods:
- Reviews
- Inspections

Dynamic test methods:
- HIL testing
- Full-scale trials

Product focus

HIL testing and process

ISDS strength: Process
LIFECYCLE SW CHANGE MANAGEMENT

MC services

SW update until OK

Identify need for SW change
- Update/patch
- Equipment change
- ...

Evaluate risk
- Possible consequences?
- Affected control systems?
- Side-effects?

Select verification method
- Based on risk, cost and nature of change

Verify SW change
- Perform chosen verification scheme
- Document verification

Install updated SW
- Onboard installation and acceptance

Ship in operation

Verification methods
- Documentation review
- Full-scale trials
- FMEA type trials
- Automatic HIL testing
- Manual HIL testing
ALL CONFUSED?
THE CHALLENGE

- Today control and safety systems are to a large degree software driven. Hence these must be included in safety management and risk assessment.

- A problem with the prevailing risk management techniques, or the use of them, is that software is either not handled at all, or handled on an abstract and too high level compared to physical components like valves, breakers and pumps.

- This is troublesome in two ways:
  - It gives a hardware biased risk assessment
  - It masks the real redundancy level of the system since “redundant” hardware components share the same software

- The end result is an incorrect risk assessment
THE SOLUTION

- ISDS
- Systems focus
- Process focus

- HIL testing / ESV
- Enhanced FMEA trials
- Verification Management

SW engineering

Testing and verification

Change management

Improved risk assessments

- SCM
- ISDS / ESV
- Maintenance systems
- Safety Management System

- SW FMEA
- SW HAZOP
- ISDS