How Realistic is the Mixed-Criticality Real-Time System Model?

Alexandre Esper, Geoffrey Nelissen, Vincent Nélis, Eduardo Tovar
Introduction

Current status

MC model gradually gaining sophistication
**Introduction**

**Current status**
- MC model gradually gaining sophistication

**Issue**
- Safety-related standards not freely accessible
  - many academic works are building on top of previous models and claims
Current status

MC model gradually gaining sophistication

Issue

Safety-related standards not freely accessible → many academic works are building on top of previous models and claims

Risk

Facilitates the propagation of misconceptions and drift from actual standards requirements
Current status

MC model gradually gaining sophistication

Issue

Safety-related standards not freely accessible
→ many academic works are building on top of previous models and claims

Risk

Facilitates the propagation of misconceptions and drift from actual standards requirements

Contribution

Elaborate on misinterpretations and discuss motivating arguments for future work
System Design and Development Assurance Process

Safety-critical Systems

- Aeronautics
- Railway
- Automotive
- Space
- Industry
System Design and Development Assurance Process

Safety-critical Systems

Aeronautics
Railway
Automotive
Space
Industry

Safety-critical Systems Development Process

System Development Process

 HW
Specs.
SW
Integ. + Test

Development Assurance Process

Safety Analyses
Reliability, Availability, Maintainability Analyses
System Design and Development Assurance Process

Safety-critical Systems

- Aeronautics
- Railway
- Automotive
- Space
- Industry

System Development Process

- HW
- SW
- Specs.
- Integ. + Test

Development Assurance Process

- Safety Analyses
- Reliability, Availability, Maintainability Analyses

System Operation

Certification Process

- Certification Authority

Redesign

CISTER - Research Center in Real-Time & Embedded Computing Systems
The Notion of Mixed-Criticality Systems

System Safety Assessment Process

- Hazard Analysis
- Fault Analysis
- Criticality Category Assignment

FMEA

<table>
<thead>
<tr>
<th>Criticality Category</th>
<th>Failure Condition Severity Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>B</td>
<td>Hazardous</td>
</tr>
<tr>
<td>C</td>
<td>Major</td>
</tr>
<tr>
<td>D</td>
<td>Minor</td>
</tr>
<tr>
<td>E</td>
<td>No Safety Effect</td>
</tr>
</tbody>
</table>

- Step 1: Define generic and function specific failure modes
- Step 2: Analyse failure causes and effects
- Step 3: Assign severities according to the established criteria
- Step 4: Identify existing compensating provisions
- Step 5: Criticality categories assignment and recommendations

Input documentation (System, Software, Operations) → Functional analysis
The Notion of Mixed-Criticality Systems

System Safety Assessment Process

Hazard Analysis ➔ Fault Analysis ➔ Criticality Category Assignment

Input documentation (System, Software, Operations) ➔ Functional analysis

Step 1: Define generic and function-specific failure modes ➔ Functions and failure modes

Step 2: Analyse failure causes and effects ➔ Failure causes and effects identified

Step 3: Assign severities according to the established criteria ➔ Severities per failure mode and failure effects

Step 4: Identify existing compensating provisions ➔ Compensating provisions

Step 5: Criticality categories assignment and recommendations ➔ Criticality classification and recommendations

Criticality Category | Failure Condition Severity Classification
---|---
A | Catastrophic
B | Hazardous
C | Major
D | Minor
E | No Safety Effect
The Notion of Mixed-Criticality Systems

System Safety Assessment Process

- Hazard Analysis
- Fault Analysis
- Criticality Category Assignment

**Criticality Category**

<table>
<thead>
<tr>
<th>Criticality Category</th>
<th>Failure Condition Severity Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>B</td>
<td>Hazardous</td>
</tr>
<tr>
<td>C</td>
<td>Major</td>
</tr>
<tr>
<td>D</td>
<td>Minor</td>
</tr>
<tr>
<td>E</td>
<td>No Safety Effect</td>
</tr>
</tbody>
</table>

**Glossary**

- **A**: Catastrophic
- **B**: Hazardous
- **C**: Major
- **D**: Minor
- **E**: No Safety Effect

**Steps**

- **Step 1**: Define generic and function specific failure modes
- **Step 2**: Analyse failure causes and effects
- **Step 3**: Assign severities according to the established criteria
- **Step 4**: Identify existing compensating provisions
- **Step 5**: Criticality categories assignment and recommendations

**Notes**

- Initial criticality assignment
- Final criticality assignment → considering compensating provisions

**Documentation**

- Input documentation (System, Software, Operations)
- Functional analysis

**Flowchart**

- Flowchart for System Safety Assessment Process

**References**

- CISTER - Research Center in Real-Time & Embedded Computing Systems
Safety-related Industrial Standards

**Automotive**
- ISO 26262: Road vehicles – Functional safety

**Railway**
- EN 50126: Railway applications – Specification and demonstration of reliability, availability, maintainability and safety
- EN 50128: Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems
- EN 50129: Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling

**Aeronautics**
- ARP 4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment
- ARP 4754: Certification Considerations for Highly-Integrated or Complex Aircraft Systems
- DO-178B/C: Software Considerations in Airborne Systems and Equipment Certification
- DO-254: Design Assurance Guidance for Airborne Electronic Hardware

**Industry**
- IEC 61508: Functional safety of E/E/PE safety-related systems
- IEC 61511: Functional safety – Safety instrumented systems for the process industry sector
- IEC 62061: Safety of machinery – Functional safety of electrical, electronic and programmable electronic control systems

**Space**
- ECSS series: Processes for project management, engineering and product assurance in space projects and applications

**Development Assurance – Safety Standards**

CISTER - Research Center in Real-Time & Embedded Computing Systems
Requirements of Safety-Related Industrial Standards

- Response timing and memory constraints
- Performance Modelling
- Resources sharing
- Partitioning
- IEC61508 (General E/E/PE)
- DO-178C (aeronautics)
- ISO26262 (automotive)
- Diverse Monitor
- Dynamic Reconfiguration
- Gracefull Degradation
- Resources sharing

IEC61508
DO-178C
ISO26262
CISTER - Research Center in Real-Time & Embedded Computing Systems
Requirements of Safety-Related Industrial Standards

Partitioning

Diverse Monitor

Dynamic Reconfiguration

IEC61508 (General E/E/PE)

DO-178C (aeronautics)

ISO26262 (automotive)

Response timing and memory constraints

Performance Modelling

Resources sharing

Graceful Degradation

Resources sharing
MCS and the Challenge of Compliance to Safety-related Standards

Safety-related Industrial Standards

...but specify stringent safety requirements

Additional challenges

No explicit MCS requirements

→ isolation and independence between applications.

Multicore + Shared Resources
MCS and the Challenge of Compliance to Safety-related Standards

Safety-related Industrial Standards

…but specify stringent safety requirements

Additional challenges

Industrial Solutions

ARINC-653 & AUTOSAR

→ isolation and independence between applications.

Multicore + Shared Resources
Most MCS work are based on Vestal model:

- Several modes of execution (1, 2, ..., L)
- Tasks → period, deadline, WCET and an assurance level
- System running in mode k
- Budget of a task is overshot
- System switches to mode k + 1
- All the tasks of criticality not greater than k are suspended (potentially reactivated)
The Use of the Word “Function”

Safety-related Industrial Standards

- Used at system level
- System functionality (HW + SW)

Academic Publications

- Used like a pure SW function
- E.g.: C function or real-time task

Academic Publications

“Function”
Mismatch of Interpretation of the Concept of “System Criticality”

Safety-related Industrial Standards

- Level of assurance (e.g. DAL, SIL, ...)
- Safety functions

Academic Publications

- Based on Vestal
- Modes of execution
- E.g. high and low criticality
Although not fundamentally wrong, it creates confusion in the context of industrial MCS

→ leads the two communities to misunderstand each others’ work
Confusion between the Notions of Criticality and Importance

Function 1

Severity: Car unusable
Probability: Probable
Controllability: Driver can keep the car on the road

Function 2

Severity: Car unusable
Probability: Probable
Controllability: Uncontrollable
Confusion between the Notions of Criticality and Importance

Function 1

Severity: Car unusable
Probability: Probable
Controllability: Driver can keep car on the road

Function 2

Severity: Car unusable
Probability: Probable
Controllability: Uncontrollable

Both are important!
Confusion between the Notions of Criticality and Importance

Function 1
Severity: Car unusable
Probability: Probable
Controllability: Driver can keep the car on the road

Function 2
Severity: Car unusable
Probability: Probable
Controllability: Uncontrollable

But ...
ASIL = Severity + Probability + Controllability

<table>
<thead>
<tr>
<th>Severity class</th>
<th>Probability class</th>
<th>Controllability class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
<td>C2</td>
</tr>
<tr>
<td>S1</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>QM</td>
</tr>
<tr>
<td>S2</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>A</td>
</tr>
<tr>
<td>S3</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>B</td>
</tr>
</tbody>
</table>
### Confusion between the Notions of Criticality and Importance

#### Function 1
- **Severity:** Car unusable
- **Probability:** Probable
- **Controllability:** Driver can keep the car on the road

#### Function 2
- **Severity:** Car unusable
- **Probability:** Probable
- **Controllability:** Uncontrollable

**But ...**

**ASIL =**
- Severity +
- Probability +
- Controllability

---

**ASIL C**

<table>
<thead>
<tr>
<th>Severity class</th>
<th>Probability class</th>
<th>Controllability class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
<td>C2</td>
</tr>
<tr>
<td><strong>S1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E2</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E3</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E4</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td><strong>S2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E2</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E3</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E4</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td><strong>S3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E2</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E3</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>E4</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>
Confusion between the Notions of Criticality and Importance

Function 1
Severity:
  Car unusable
Probability:
  Probable
Controllability:
  Driver can keep the car on the road

Function 2
Severity:
  Car unusable
Probability:
  Probable
Controllability:
  Uncontrollable

But ...
ASIL = Severity + Probability + Controllability

<table>
<thead>
<tr>
<th>Severity class</th>
<th>Probability class</th>
<th>Controllability class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
<td>C2</td>
</tr>
<tr>
<td>S1</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>QM</td>
</tr>
<tr>
<td>S2</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>A</td>
</tr>
<tr>
<td>S3</td>
<td>E1</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E2</td>
<td>QM</td>
</tr>
<tr>
<td></td>
<td>E3</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>E4</td>
<td>B</td>
</tr>
</tbody>
</table>

ASIL C
ASIL D
Confusion between the Notions of Criticality and Importance

Function 1

Severity:
Car unusable

Probability:
Probable

Controllability:
Driver can keep the car on the road

Function 2

Severity:
Car unusable

Probability:
Probable

Controllability:
Uncontrollable

We cannot always stop lower criticality tasks in favour of higher criticality ones
Confusion between the Notions of Criticality and Importance

Example from IEC61508

System Control

Error and Event Logging

Sends Data

HW A

HW B
Confusion between the Notions of Criticality and Importance

Example from IEC61508

Highest criticality

System Control

Error and Event Logging

Sends Data

HW A

HW B
Confusion between the Notions of Criticality and Importance

Example from IEC61508

- Highest criticality
  - System Control
  - Error and Event Logging

Inherits criticality

Sends Data

HW A

HW B
Confusion between the Notions of Criticality and Importance

Example from IEC61508

- Highest criticality
  - System Control
  - Error and Event Logging
  - Sends Data
- Hardware failure
  - HW A
  - HW B

Example from IEC61508
Confusion between the Notions of Criticality and Importance

Example from IEC61508

- **Highest criticality**
  - System Control

- **Stop**
  - Error and Event Logging

- **Hardware failure**
  - HW A

- **Highest criticality**
  - HW B
Confusion between the Notions of Criticality and Importance

Example from IEC61508

We can sometimes stop high criticality tasks

Hardware failure

System

Highest criticality

Error and Event Logging

Highest criticality

HW A

HW B

CISTER - Research Center in Real-Time & Embedded Computing Systems
Vestal’s model:
High and Low criticality tasks run on the same processor and scheduler
Vestal’s model:
High and Low criticality tasks run on the same processor and scheduler

Industrial perspective

Failure mode:
Period change of external event

FMEA analysis

Processor A

LC task

HC task
Vestal’s model:
High and Low criticality tasks run on the same processor and scheduler

Industrial perspective

Failure mode:
Period change of external event

Local Effect:
Change of LC task period

FMEA analysis

Processor A

HC task

LC task

CISTER - Research Center in Real-Time & Embedded Computing Systems
Vestal’s model: High and Low criticality tasks run on the same processor and scheduler

Industrial perspective

Failure mode: Period change of external event

Local Effect: Change of LC task period

FMEA analysis

Processor A

HC task

LC task
Vestal’s model:
High and Low criticality tasks run on the same processor and scheduler

Industrial perspective

Failure mode:
Period change of external event

Local Effect:
Change of LC task period

End Effect:
HC task misses deadline

Processor A

FMEA analysis
Vestal’s model:
High and Low criticality tasks run on the same processor and scheduler

Industrial perspective

Failure mode:
Period change of external event

Local Effect:
Change of LC task period

Processor A

FMEA analysis
Inherits criticality

End Effect:
HC task misses deadline

CISTER - Research Center in Real-Time & Embedded Computing Systems
Vestal’s model:
High and Low criticality tasks run on the same processor and scheduler

Industrial perspective

Failure mode:
Period change of external event

FMEA analysis

Local Effect:
Change of LC task period

End Effect:
HC task misses deadline

HC and LC tasks not isolated in time all tasks will have to be certified at HC level
Vestal’s model:
High and Low criticality tasks run on the same processor and scheduler

- Will never be able to convince a certification authority that the tasks are isolated in time
- The cost of the system would increase exponentially...
- We miss the initial goal of integrating a mixed-criticality system in the same platform to decrease costs
WCET Estimation

Vestal’s model & Derivatives:
Assumption: Higher degree of assurance of a task $\rightarrow$ more pessimistic WCET estimation

- WCET upperbound $\rightarrow$ necessary but not sufficient condition to ensure the safety
Vestal’s model & Derivatives:
Assumption: Higher degree of assurance of a task → more pessimistic WCET estimation

- WCET upperbound → necessary but not sufficient condition to ensure the safety
- Requires mechanisms to ensure that safety is not compromised in case of timing violation
  - E.g. time partitioning
Probabilistic WCET
- Provides a probabilistic upper-bound on the execution time
- Aims at building a reliability model of the software
Probabilistic WCET
- Provides a probabilistic upper-bound on the execution time
- Aims at building a reliability model of the software

Software reliability models:
- Still under debate
- Confidence cannot be placed in such models
Probabilistic WCET
- Provides a probabilistic upper-bound on the execution time
- Aims at building a reliability model of the software

Software reliability models:
- Still under debate
- Confidence cannot be placed in such models

- Important research direction
- But...cannot assume that they will ever be used in industrial systems to prove software safety
Probabilistic WCET
- Provides a probabilistic upper-bound on the execution time
- Aims at building a reliability model of the software

Software reliability models:
• Still under debate
• Confidence cannot be placed in such models

• Important research direction
• But...cannot assume that they will ever be used in industrial systems to prove software safety

• Need to work on the safety argumentation
Probabilistic WCET
- Provides a probabilistic upper-bound on the execution time
- Aimed at building a reliability model of the software
Typical question: would you fly an airplane designed with probabilistic software reliability models?

Probabilistic WCET
- Provides a probabilistic upper-bound on the execution time
- Aimed at building a reliability model of the software
Probabilistic WCET
- Provides a probabilistic upper-bound on the execution time
- Aimed at building a reliability model of the software

Typical question: would you fly an airplane designed with probabilistic software reliability models?
Conclusion

• Clear gap between some of the guidelines provided in safety-related standards and their interpretation by the academic community

• Misalignment of terminology leads to misunderstanding of each other’s work

• Confusion between the notions of criticality and importance

• Ensuring safety in terms of timing isolation goes beyond accurate WCET estimates

• Probabilistic WCET estimates: in case that direction is followed → need to work on the argumentation
Question & Answers