APPLYING SAFETY AND SYSTEMS ENGINEERING PRINCIPLES FOR ANTIFRAGILITY

Eric Verhulst, CEO/CTO
Altreonic NV
Content

- Safety engineering and Safety Integrity Levels (SIL)
- Some issues with the SIL criterion
- Introducing the normative ARRL criterion
- Illustrated architectures
- ARRL and antifragility
- Autonomous traffic and ARRL-7
- Conclusions
- Note: Work In Progress!
Systems Engineering vs. Safety Engineering

- System = holistic
- Real goal is "Trustworthy Systems"
  - Cfr. Felix Baumgartner almost did not do it because he didn't trust his safe jumpsuit
- TRUST = by the user or stakeholders
  - Achieving intended Functionality
  - Safety & Security & Usability & Privacy
  - Meeting non-functional objectives
    - Cost, energy, volume, maintainability, scalability, Manufacturability,..
- So why this focus on safety?
- User expects guaranteed “QoS” from a “Trustworthy system”
Safety and certification

- **Safety** can be defined to be the control of **recognized hazards** to achieve an **acceptable level of risk**.
  - Safety is general property of a system, not 100% assured
  - It is complex but there are moral liabilities
- **Certification:** In depth review => safe to operate
  - “Conformity assessment” (for automotive)
  - Not a technical requirement: confidence, legal
- **Evidence makes the difference:**
  - Evidence is a **coherent** collection of **information** that relying on a number of **process artifacts** linked together by their **dependencies and sufficient structured arguments** provides an **acceptable proof** that a specific system goal has been reached.
Categorisation of Safety Risks

<table>
<thead>
<tr>
<th>Category</th>
<th>Consequence upon failure</th>
<th>Typical SIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catastrophic</td>
<td>Loss of multiple lives</td>
<td>4</td>
</tr>
<tr>
<td>Critical</td>
<td>Loss of a single life</td>
<td>3</td>
</tr>
<tr>
<td>Marginal</td>
<td>Major injuries to one or more persons</td>
<td>2</td>
</tr>
<tr>
<td>Negligible</td>
<td>Minor injuries at worst or material damage</td>
<td>1</td>
</tr>
<tr>
<td>No consequence</td>
<td>No damages, user dissatisfaction</td>
<td>0</td>
</tr>
</tbody>
</table>

- $\text{SIL} \equiv f(\text{probability of occurrence, severity, controllability})$
- As determined by HARA
- SIL goals $\equiv$ Risk Reduction Factor
- Criteria and classification are open to interpretation
Problems with SIL definition

- Poor harmonization of definition across the different standards bodies which utilize SIL => Reuse?
- Process-oriented metrics for derivation of SIL
- SIL level determines architecture (system specific)
- Estimation of SIL based on reliability estimates
  - System complexity, particularly in software systems, makes SIL estimation difficult if not impossible
  - based on probabilities that are very hard if not impossible to measure and estimate
  - Reliability of software (discrete domain) is not statistical!
  - The law of Murphy still applies:
    - The next instant can be catastrophic
New definition: start from the component up

**ARRL:** Assured Reliability and Resilience Level

<table>
<thead>
<tr>
<th>ARRL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>it might work (use as is)</td>
</tr>
<tr>
<td>1</td>
<td>works as tested, but no guarantee</td>
</tr>
<tr>
<td>2</td>
<td>works correctly, IF no fault occurs, guaranteed no errors in implementation) =&gt; formal evidence</td>
</tr>
<tr>
<td>3</td>
<td>ARRL 2 + goes to fail-safe or reduced operational mode upon fault (requires monitoring + redundancy) - fault behavior is predictable as well as next state</td>
</tr>
<tr>
<td>4</td>
<td>ARRL 3 + tolerates one major failure and is fault tolerant (fault behavior predictable and transparent for the external world). Transient faults are masked out</td>
</tr>
<tr>
<td>5</td>
<td>The component is using heterogeneous sub-components to handle residual common mode failures</td>
</tr>
</tbody>
</table>
ARRL: what does it mean?

- **Assured:**
  - There is verified, trustworthy evidence
  - Process related and architecture related

- **Reliability:**
  - In absence of faults, MTBF is >> life-time: QA aspects

- **Resilience:**
  - The fault behaviour is predicted: trustworthy behaviour
  - Capability to continue to provide core function

- **Level: ARRL is normative**
  - Components can be classified: contract
Consequences

• If a system/component has a fault, it drops into a degraded mode => lower ARRL
  • ARRL3 is the operational mode after an ARRL4 failure
    • Functionality is preserved
    • Assurance level is lowered

• SIL not affected and domain independent
  • System + environment + operator defines SIL

• ARRL is a normative criterion:
  • Fault behavior is made explicit: verifiable
  • Cfr. IP-norm (comes with a predefined test procedure)
ARRL-3
Unanticipated input values

\[
\begin{align*}
\{ & i_1, i_2, \ldots, i_3, \ldots, i_n \} \\
\end{align*}
\]

Input/output guards:
Guaranteed bounded

\[
\begin{align*}
\{ & o_1, o_2, \ldots, o_3, \ldots, o_n \} \\
\end{align*}
\]

Common mode failures possible
Induced fault
One or more state space trees:
Monitor and supervisor sub-component
Fail safe output

Comparator
N out of M voter/demux
ARRL-4

N out of M voter/mux
ARRL-4

Output preserved

Process related common mode failures minimised

(t)
SIL and ARRL are complementary
A system is never alone
What means “anti-fragile”?

• New term quoted by Taleb

• An anti-fragile system gets “better” after being exposed to “stressors”
  • Better: we need a metric => QoS?
  • Stressors: cfr. hazard, faults, ...
  • The issue in safety: rare events (improbable a priori, certain post factum) (Taleb’s “black swan”)

• What does it mean in the context of safety/systems engineering? Isn’t ARRL-5 not the top level?
Two example domains

- **Automotive:**
  - 1,2 million people killed/year: **daily event**
  - Cars get better, but people get killed: safer? QoS?

- **Aviation:**
  - 500 people killed/year: **a rare event**
  - Planes get better, cheaper, safer, energy-efficient

- **Railway, telecommunications, medical, ...**
  - Similar examples

- **What sets them apart?**
Assessment in terms of ARRL

**Automotive:**
- Vehicle is an ARRL-3 system
- Upon fault, presumed to go the fail-safe state
- No black box, no records, ...
- Automotive is a collection of vehicles

**Aviation:**
- Planes are ARRL-5
- Upon fault, redundancy takes over
- Black box, central database,
- Preventive maintenance
- Aviation is an eco-system
Extended systems (of systems) view
Preconditions for anti-fragility

- **Extensive domain knowledge**: experience
- **Openness**: shared critical information
- **Feedback loops** at several levels between large number of stakeholders
- **Independent supervision**: guidance
- Core components are **ARRL-4 or -5**
- **The system is the domain**
- **Service matters more** than the component
### ARRL-6 and ARRL-7 (inherits ARRL-5)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ARRL 3</strong></td>
<td>ARRL 2 + goes to fail-safe or reduced operational mode upon fault (requires monitoring + redundancy) - fault behavior is predictable as well as next state</td>
</tr>
<tr>
<td><strong>ARRL 4</strong></td>
<td>ARRL 3 + tolerates one major failure and is fault tolerant (fault behavior predictable and transparent for the external world). Transient faults are masked out</td>
</tr>
<tr>
<td><strong>ARRL 5</strong></td>
<td>The component is using heterogeneous sub-components to handle residual common mode failures</td>
</tr>
<tr>
<td><strong>ARRL 6</strong></td>
<td>The component (subsystem) is monitored and a <strong>process</strong> is in place that maintains the system’s functionality</td>
</tr>
<tr>
<td><strong>ARRL 7</strong></td>
<td>The component (subsystem) is art of a <strong>system of systems</strong> and a <strong>process is in place</strong> that includes continuous monitoring and improvement supervised by an <strong>independent regulatory body</strong></td>
</tr>
</tbody>
</table>
Autonomous traffic

- Self-driving cars are the future? Cfr. Google car
- Systems engineering challenge much higher than flying airplanes
- Huge impact: socio-economic “black swan”
- Pre-conditions:
  - Vehicles become ARRL-5
  - System = traffic, includes road infrastructure
  - Standardisation (vehicles communicate)
  - Continuous improvement process
- Hence: needs ARRL-7
Beyond ARRL-7

• Not all systems are engineered by humans

• Biological systems:
  • Survivability (selection) and adaption
  • Build-in mechanism (long term feedback loops)
  • ARRL-8 ?
  • Inheritance of ARRL-7 ?

• Genetic engineering:
  • Directed selection and adaptation
  • ARRL-9? Or ARRL-7 with bio-components?
Conclusions

- ARRL concept allows compositional safety engineering with reuse of components/subsystems
- More complex systems can be safer
- A unified ARRL aware process pattern can unify systems and safety engineering standards
- ARRL-6 and ARRL-7 introduce a system that include a feedback loop process during development but also during operation

- ANTIFRAGILE = ARRL-7

More info:
www.altreonic.com
Further work

• Making ARRL normative and applicable
  • Refinement and Completeness of criteria
  • Normative: components carry contract and evidence
    • Independent of final use or application domain
    • Process evidence + validated properties
    • ARRL-3 and higher: HW/SW co-design?
  • Study link with a system’s critical states
  • Apply it on real cases

• Input and feedback welcome