Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2014) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

"1st International Workshop "From Dependable to Resilient, from Resilient to Antifragile Ambients and Systems" (ANTIFRAGILE 2014)"

Applying systems and safety engineering principles for antifragility

Eric Verhulst[a]*

[a]*Altreonic NV, Gemeentestraat 61AB1, B3210 Linden, Belgium*

Abstract

Traditional systems engineering can be seen as a range of activities and techniques aiming at developing a system (or product) in a systematic and predictable way. In particular safety engineering standards aim at developing "safe" systems that continue to provide their intended behaviour even when faults or hazards occur. Often, the approach will be relatively static and the aim will be to mitigate the analysed risks as this is the easiest way to verify the safety property. Antifragile systems [10] however are defined as systems that adapt after a fault occurs and are not only resilient but also learn from faults and incidents to improve their delivered service level. This paper investigates antifragility in the context of systems engineering and proposes a normative criterion that helps to understand the pre-conditions reaching antifragility.

Selection and peer-review under responsibility of Elhadi M. Shakshuki.

Keywords: antifragility, safety engineering, systems engineering, safety integrity level, assured realiability and resilience level, self adaptive systems,

* Corresponding author. Eric Verhulst, Tel.:+32 16 202059
E-mail address: eric.verhulst@altreonic.com

Selection and peer-review under responsibility of Elhadi M. Shakshuki.

1 Introduction

Systems engineering aims at developing systems that meet the requirements and constraints of its stakeholders. Increasingly systems must not only provide their intended functionality, but it must also be guaranteed in a certifiable way that such systems remain safe (and secure) when subjected to faults or hazardous situations. From the safety point of view, the lower the required residual risks should be, the higher the safety related requirements, often expressed as SIL (Safety Integrity Levels). The same applies for the subsystems whose faults can induce a safety risk.

We have argued before [1,2,3] that this view is rather narrow. In reality what matters is how much the stakeholders (including the users) consider the system as trustworthy whereby safety is one of the specified properties. Similarly, what a user expects is a guaranteed QoS (Quality of Service) level. Depending on the level, it guarantees that the system will be able to deliver its intended functionality even if faults occur. Hence, the ultimate case is one where the system survives faults. As this criterion is very wide, this led to the introduction of a novel more normative criterion, called ARRL (Assured Reliability and Resilience Level) that differentiates between the failure conditions and how the system coops with it.

Depending on the severity of the fault scenario and the desired continuity of the system's functions this requires increasingly higher levels of ARRL. In traditional systems engineering, the continuation of the services is achieved by reconfiguring the architecture and by redundancy. The question is whether this is sufficient or a necessary condition to reach the novel property of antifragility [10]. Before we answer the question, we recapitulate the existing notions of SIL, QoS and ARRL.

2 Concise overview of existing criteria in the domain of trustworthiness

2.1 Safety Integrity Level

We consider first the IEC 61508 standard [4], as this standard is relatively generic. It considers mainly programmable electronic systems. The goal is to bring the risks to an acceptable level by applying safety functions. IEC 61508 starts from the principle that safety is never absolute; hence it considers the likelihood of a hazard (a situation posing a safety risk) and the severity of the consequences. A third element is the controllability. The combination of these three factors is used to determine a required SIL, categorized in 4 levels, SIL-1 being the lowest and SIL-4 being the highest. These levels correspond with normative allowed Probabilities of Failure per Hour and require corresponding Risk Reduction Factors that depend on the usage pattern (infrequent versus continuous). The risk reduction itself is achieved by a combination of reliability measures (higher quality), functional measures as well as assurance from following a more rigorous engineering process. The safety risks are generally classified in 4 classes, roughly each corresponding with a required SIL level whereby we added a SIL-0 for completeness. It must be said however that the standards allow quite some room for interpretation, in particular when it comes to the use of probabilities and assessment of the controllability factor.

Table 1 Categorisation of Safety Risks

Category	Typical SIL	Consequence upon failure
Catastrophic	4	Loss of multiple lives
Critical	3	Loss of a single life
Marginal	2	Major injuries to one or more persons
Negligible	1	Minor injuries at worst or material damage only
No consequence	0	No damages, except user dissatisfaction

The SIL level is used as a directive to guide selecting the required architectural support and development process requirements. For example SIL-4 imposes redundancy and positions the use of formal methods as highly recommended.

2.2 Quality of Service Levels

A system that is being developed is part of a larger system that includes the user (or operator) as well as the environment in which the system is used. Note as well that this is a hierarchical notion. A system can be a subsystem or a component in a large system and can also include services and processes that support the final mission of a system.

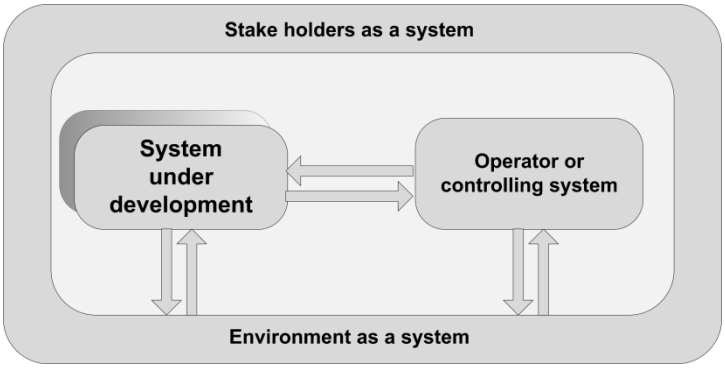

Figure 1 The context in which a system is developed and used

From the user's point of view, the system must deliver an acceptable and predictable level of service, which we call the Quality of Service (QoS). A failure in a system is not seen as an immediate risk but rather as a breach of contract on the QoS whereby the system's malfunction can then result in a safety related hazard or loss of mission control, even when no safety risks are present. As such we can see that a given SIL is a subset of the QoS. The QoS can be seen as the availability of the system as a resource that allows the user's expectations to be met. Aiming to reduce the intrinsic ambiguities of the Safety Levels we now formulate a scale of QoS as follows:

Table 2. Quality of Service levels

QoS	There is no guarantee that there will be resources to sustain the service. Hence the user should not rely on the system and should consider it as untrustworthy. When using the system, the user is taking a risk that is not predictable.
QoS-2	The system must assure the availability of the resources in a statistically acceptable way. Hence, the user can trust the system but knows that the QoS will be lower from time to time. The user's risk is mostly one of annoyance and dissatisfaction or of reduced service.
QoS-3	The system can always be trusted to have enough resources to deliver the highest QoS at all times. The user's risk is considered to be negligible.

The classification leaves room for residual risks but those are not considered design goals but rather as uncontrollable risks. Neither the user nor the system designer has much control over them. This is due to the existence of non-linear discrete subsystems (mainly digital electronics and software) which was elaborated further in [5]. This aspect will be important when we discuss antifragility further in this text.

2.3 The ARRL criterion

We introduce the ARRL or Assured Reliability and Resilience Level to guide us in composing safe and trustworthy systems. The different ARRL classes are defined in Table 3. They are mainly differentiated in terms of how much assurance they provide in meeting their contract in the presence of faults. The reader should keep in mind that the term component can also be a (sub)-system or system acting as components in a larger system.

Table 3 ARRL Levels

	ARRL definition
Inheritance property	Each ARRL level inherits all properties of any lower ARRL level.
ARRL-0	The component might work ("use as is"), but there is no assurance. Hence all risks are with the user.
ARRL-1	The component works "as tested", but no assurance is provided for the absence of any remaining issues.
ARRL-2	The component meets all its specifications, if no fault occurs. This means that it is guaranteed that the component has no implementation errors, which requires formal evidence as testing can only uncover testable cases. The formal evidence does not necessarily provide complete coverage but should uncover all so-called systematic faults, e.g., a wrong parameter value. In addition, the component can still fail due to randomly induced faults, for example an externally induced bit-flip.
ARRL-3	The component is guaranteed to reach a fail-safe or reduced operational mode upon a fault. This requires monitoring support and some form of architectural redundancy. Formally speaking this means that the fault behavior is predictable as well as the subsequent state after a fault occurs. This implies that specifications include all fault cases as well as how the component should deal with them.
ARRL-4	The component can tolerate one major fault. This corresponds to requiring a fault-tolerant design. This entails that the fault behavior is predictable and transparent to the external world. Transient faults are masked out.
ARRL-5	The component is using heterogeneous sub-components to handle residual common mode failures.

We should mention that there is an implicit assumption about a system's architecture when defining ARLL. A system is composed by defining a set of interacting components. This has important consequences:

1. The component must be designed to prevent the propagation of errors. Therefore the interfaces must be clearly identifiable and designed with a "guard". These interfaces must also be the only way a component can interact with other components. The internal state is not accessible from another component, but can only be made available through a well-defined protocol (e.g. whereby a copy of the state is communicated).
2. The interaction mechanism, for example a network connection, must carry at least the same ARRL credentials as the components it interconnects. Actually, in many cases, the ARLL level must be higher if one needs to maintain a sufficiently high ARRL level at the level of the (sub)-system composed of the components.
3. Hence, it is better to consider the interface as a component on itself, rather than for example assuming an implicit communication between the components.

3 Is this sufficient for antifragility?

The normative ARRL levels describe as the name says, levels of reliability and resilience. They approach the notion of graceful degradation by redundancy but assuming that in absence of faults the system components can be considered as error-free. The additional functionality and redundancy (that is also error-free) is to be seen as an architectural or process level improvement. But in all cases, contrary to the antifragility notion, the system will not gain in resilience or reliability. It can merely postpone catastrophic failures while maintaining temporally the intended services. It does this by assuming that all types of faults can be anticipated, which would be the state of the art in engineering.

3.1 Antifragility assumptions

However, the proposed scheme introduces already two concepts that are essential to take it a step further. Firstly, there is redundancy in architecture and process and secondly, there is a monitoring function that acts by reconfiguring the system upon detecting a fault.

So, how can a system become "better" when subjected to faults? As we introduce a metric as a goal, we must somehow measure and introduce feedback loops. If we extrapolate and scale up, this assumes that the system has a type of self-model of itself that it can use to compare its current status with a reference goal. Hence, either the designer must encapsulate this model within the system or the model is external and becomes part of the system. If we consider systems that include their self-model from the start, then clearly becoming a "better" system has its limits, the limit being the designers's idea at the moment of conception. While there are systems that evolve to reach a better optimum (think about neural networks or genetic algorithms), these systems evolve towards a limit value. In other words they do not evolve, they converge.

If on the other hand we expand the system as in Figure 1, then the system can evolve. It can evolve and improve because we consider its environment and all its stakeholders of which the users as part of the system. They continuously provide information on the system's performance and take measures to improve upon it. It also means that the engineering process doesn't stop when the system has been put to use for the first time. It actually never ends because the experience is transferred to newer designs.

There are numerous examples of antifragile systems already at work, perhaps not perfect all the time though most of the time. A prime example is the aviation industry that demonstrates by its yearly decreasing number of fatalities and quality of service that it meets the criterion of antifragility. Moreover, it is a commercial success. So let's examine some of its properties and extract the general principles, as reflected in the aviation standards and practice [6].

Table 4 Generic properties derived from observing the avionic sector

Aviation specific	**Generic property**
The industry has a long track record	The domain has undergone many technological changes whereby an extensive knowledge was built up.
Development of systems follows a rigorous, quantifiable, certifiable process, that is widely published and adopted.	The process is open and reflects the past experience and is certified by an independent external authority.
Certification requirements foster developing "minimal" implementations that still meet the operational requirements.	Systems are designed to be transparent and simple, focusing on the must-haves and not on the nice to haves.
Airplanes are designed to be 100% safe and to be operated in 100% safe conditions.	The domain has a goal of perfection. Any deviation is considered a failure that must be corrected. By design the system, its components and operating procedures aim at absence of service and safety degradation.
Any failure is reported in a public database and thoroughly analysed.	Any issue is seen as a valuable source of information to improve processes and systems.
Airplanes are operated as part of a larger worldwide system that involves legal authorities, the operators, the manufactures, the public and supervising independent authorities.	A (sub)system is not seen in isolation but in its complete socio-economic context. This larger system is self-regulating but supervised and controlled by an independent authority.
Airplanes have a long life time and undergo mid-life updates to maintain their serviceability	The focus is on the service delivered and not on the system as a final product.
Fault conditions are preventively monitored. The system is fault tolerant through redundancy, immediate repair and preventive maintenance.	A process is in place that maintains the state of the system at a high service level without disrupting the services provided.

3.2 Some industries are antifragile by design

To remain synoptic, we will list a few key principles of the aviation industry and derive from them key generic principles which apply to other systems and provide them with antifragile properties.

Table 4 can also be related to many other domains that have a significant societal importance. Think about sectors like medical devices, railway, automotive, telecommunications, internet, nuclear, etc. They all have formalized safety standards which must be adhered to because when failing they have a high impact at socio-economic level.

At the same time, systems like railway that are confined by national regulations clearly have a higher challenge to continue delivering their services at a high level. As a counter example we can take a look at the automotive sector. Many more people are killed yearly in traffic than in airplanes, even if cars today are stuffed with safety functions. In the next section we will explore this more in detail.

Deducting some general properties out of the table 4, we can see that systems that could be termed antifragile are first of all not new. Many systems have antifragile properties. Often they can be considered as complex (as there are many components in the system) but they remain resilient and antifragile by adopting a few fundamental rules:

1. Openness: all service critical information is shared and public.
2. Constant feedback loops between all stakeholders at several different levels.
3. Independent supervising authorities.
4. The core components are designed at ARRL-4 and ARRL-5 levels, i.e. fault tolerant.

3.3 Do we need an ARRL-6 and ARRL-7 level?

An ARRL-5 system can be seen as a weak version of a resilient system. While it can survive a major fault, it does so by dropping into an ARRL-4 mode. The next failure is likely catastrophic. However airplanes are also designed as part of a larger system that helps to prevent reaching that state. Continuous build-in-test functions and diagnostics will detect failures before they become a serious issue. Ground crews will be alerted over radio and will be ready to replace the defective part upon arrival at the next airport. We could call this the ARRL-6 level whereby fault escalation is constrained by early diagnostics and monitoring and the presence of a repair process that maintains the operational status at an optimal level. Note that in large systems like server farms and telecommunication networks similar techniques are used. Using monitoring functions and hot-swap capability on each of the 1000's of processing nodes, such a system can reach almost an infinite lifetime (economically speaking). Even the technology can be upgraded without having to shut down the system.

The latter example points us in the direction of what a normative ARRL-7 level could be. It is a level whereby the system is seen as a component in a larger system that includes a continuous monitoring and improvement process. The later implies a learning process as well. The aviation industry seems to have reached this maturity level. The term maturity is no coincidence, it reminds of the maturity levels as defined by CMMI levels for an organisation. Table 5 summarizes the new ARRL levels whereby we remind the reader that each ARRL level inherits the properties of the lower ARRL levels.

Table 5 ARRL-6 and ARRL-7 definitions

ARRL-6	The component (or subsystem) is monitored and designed for preventive maintenance whereby a supporting process repairs or replaces defective items while maintaining the functionality and system's services.
ARRL-7	The component (or subsystem) is part of a larger "system of systems" that includes a continuous monitoring and improvement process supervised by an independent regulating body.

4 Automated traffic as an antifragile ARRL-7 system

As we discussed earlier [1, 2, 3, 5], the automotive sector does not yet meet the highest ARRL levels as well as in the safety standards (like IEC-26262) [7] and in reality (1000 more people are killed in cars than in airplanes worldwide and even a larger number survive with disabilities)[8,9,11]. The main reason is not that cars are unsafe by design (although fault tolerance is not supported) but because the vehicles are part of a much larger traffic system

that is largely an ad-hoc system. Would it be feasible to reach a similar ARRL level as in the aviation industry? What needs to change? Can this be done by allowing autonomous driving?
A first observation is that the vehicle as a component now needs to reach ARRL-4, even ARRL-5 and ARRL-6 levels. If we automate traffic, then following design parameters become crucial:

- The margin for driving errors will greatly decrease. Vehicles already operate in very dynamic conditions whereby seconds and centimetres make the difference between an accident and not an accident. With automated driving, bumper to bumper driving at high speed will likely be the norm.
- The driver might be a back-up solution to take over when systems fail, but he is unlikely to be well enough trained and therefore to react in time (seconds).
- A failing vehicle can generate a serious avalanche effect whereby many vehicles become involved and the traffic system can be seriously disrupted.

Hence, vehicles need to be fault tolerant. First of all they constantly monitor and diagnose the vehicle components to prevent pro-actively the failing of subsystems and secondly when a failure occurs the function must be maintained allowing to apply repair in a short interval.
A second observation is that the automated vehicle will likely constantly communicate with other vehicles and with the traffic infrastructure. New vehicles start to have this capability today as well, but with automated vehicles this functionality must be guaranteed at all times as disruption of the service can be catastrophic.
A third observation is that the current road infrastructure is likely too complex to allow automated driving in an economical way. While research vehicles have been demonstrated the capability to drive on unplanned complex roads, the question is whether this is the most economical and trustworthy solution.
Automated traffic can be analysed in a deeper way. Most likely, worldwide standardisation will be needed and more openness on when things fail. Most likely, fully automated driving providing very dense traffic at high speed will require dedicated highways, whereas on secondary roads the system will be more a planning and obstacle avoidance assistant to the driver. One can even ask if we should still speak of vehicles. The final functionality is mobility and transport. For the next generation, cars and trucks as we know them today might not be the solution. A much more modular and scalable, yet automated, transport module that can operate off-road and on standardised auto-highways is more likely the outcome. Users will likely not own such a module but rent it when needed whereby operators will be responsible for keeping it functioning and improving it without disrupting the service. Independent authorities will supervise and provide an even playing field. Openness, communication and feedback loops at all levels will give it the antifragility property that we already enjoy in aviation.

5 Is there an ARRL-8 level and higher?

One can ask the question whether we can define additional ARRL levels. ARRL levels 0 to 7 are clearly defined in the context of (traditional) systems engineering whereby humans are important agents in the required processes to reach these levels. One could say that such a system as shown in Figure 1 is self-adaptive. However the antifragile properties (even when only partially fulfilled) are designed in and require conscious and deliberate actions to maintain the ARRL level. If we look at biological systems we can see that such systems evolve without the intervention of external agents (except when they stress the biological system). Evolution as such has reached a level whereby the “architecture” is self-adaptive and redundant without the need for conscious and deliberate actions. We could call this the ARRL-8 level.
When considering bio- and genetic engineering, we can see that we could take it a step further. Genetic engineering (and that includes early breeding techniques) involves human intervention in ARRL-8 level systems. The boundaries however become fuzzy. One could consider this as an ARRL-9 level but also as an ARRL-7 level using biological components. This raises interesting philosophical and ethical questions that requires a deeper understanding on how genetic building blocks really work. This topic requires further study and is not within the scope of this paper.

6 Conclusions

Taled [10] defines antifragile mostly in the context of a subjective human social context. He quotes the term to indicate something beyond robustness and resilience that reacts to stressors (and alike) by actually improving its resistance to such stressors. Taking this view in the context of systems engineering we see that such systems already

exist. They are distinguished by considering the system as a component in a greater system that includes the operating environment and it continuous processes and all its stakeholders. Further differences are a culture of openness, continuous striving for perfection and the existence of numerous multi-level feedback loops whereby independent authorities guide and steer the system as a whole. The result is a system that evolves towards higher degrees of antifragility. An essential difference with traditional engineering is that the system is continuously being redefined and adapted in an antifragile process.

The study has allowed us to define two new levels for the normative ARRL criterion, ARRL-6 indicating a system that preventively seeks to avoid failures by repair and ARRL-7 whereby a larger process is capable of not only repairing but also updating the system in a controlled way without disrupting its intended services. Given the existence of systems with such (partial) properties, it is not clear whether the use of the neologism “antifragile” is justified to replace reliability and resilience, even if it indicates a clear qualitative and distinctive level. This will need further study.

Acknowledgements

With thanks to Vance Hilderman for his comments and review.

References

[1] Eric Verhulst, Bernhard Sputh, Jose Luis de la Vara, Vincenzo de Florio, ARRL: a novel criterion for Composable Safety and Systems Engineering. SafeComp/SASSUR workshop. Toulouse, September 2013.

[2] Eric Verhulst, Bernhard Sputh, Jose Luis de la Vara, Vincenzo de Florio. From Safety Integrity Level to Assured Reliability and Resilience Level for composable safety critical systems, ICSSEA, Paris, Nov. 2013.

[3] Eric Verhulst, Bernhard Sputh. ARRL, a criterion for compositional safety and systems engineering. A normative approach to specifying components. IEEE ISRRE2013, Pasedena, November 2013.

[4] http://www.iec.ch/functionalsafety/. Functional safety of electrical / electronic / programmable electronic safety-related systems (IEC 61508) (2005)

[5] http://www.altreonic.com/sites/default/files/Altreonic_ARRL_DRAFT_WIP011113.pdf. From Safety Integrity Level to Assured Reliability and Resilience Level for Compositional Safety Critical Systems (internal white paper)

[6] http://www.rtca.org

[7] IEC: ISO: International Standard Road vehicles - Functional safety - ISO/DIS 26262 (2011)

[8] BAA: Aircraft Crashes Record Office. http://baaa-acro.com/index.html (2013)

[9] World Health Organization: WHO global status report on road safety 2013: supporting a decade of action. Technical Report (2013)

[10] Antifragile. Things that gain from disorder. Nassim Nicholas Taleb. Random House (Nov. 2012)

[11] http://ec.europa.eu/environment/noise/home.htm