# Safe and secure software updates on high-performance embedded systems

Irune Agirre
*Ikerlan Technology Research Centre*
*Basque Research and Technology Alliance (BRTA)*
*Arrasate, Spain*
*iagirre@ikerlan.es*

***Abstract*—The next generation of dependable embedded systems feature autonomy and higher levels of interconnection. Autonomy is commonly achieved with the support of artificial intelligence algorithms that pose high computing demands on the hardware platform, reaching a high performance scale. This involves a dramatic increase in software and hardware complexity, fact that together with the novelty of the technology, raises serious concerns regarding system dependability. Traditional approaches for certification require to demonstrate that the system will be acceptably safe to operate before it is deployed into service. The nature of autonomous systems, with potentially infinite scenarios, configurations and unanticipated interactions, makes it increasingly difficult to support such claim at design time. In this context, the extended networking technologies can be exploited to collect post-deployment evidence that serve to oversee whether safety assumptions are preserved during operation and to continuously improve the system through regular software updates. These software updates are not only convenient for critical bug fixing but also necessary for keeping the interconnected system resilient against security threats. However, such approach requires a recondition of the traditional certification practices.**

***Keywords*-OTASU, safety, security, autonomous systems**

## I. Extended Abstract

Dependable embedded systems in which safety is a critical factor are subject to certification. Certification is the judgment that a given system is suitable and safe enough for its intended use, where safe enough refers to the absence of unacceptable risks leading to catastrophic consequences caused by the malfunctioning of the embedded system. One of the most common practices to achieve such certification is proving adherence to standards. Standards define prescribed processes for system development and require the adoption of specific techniques that aim to reduce the probability of failure down to an acceptable threshold, which is associated to the safety integrity or criticality level of the system. There is an extensive set of functional safety standards to aid in the certification process. Despite most standards target domain-specific applications (like ISO 26262 [8] for automotive or IEC 50128 [4] for railway), many of them are based on the generic international IEC 61508 [6] standard for the Functional Safety of Electrical/Electronic/Programmable Electronic (E/E/PE) safety-related systems, which is applicable across multiple industrial sectors.

The stringent requirements posed by standards have been successfully applied in relatively simple, predictable and isolated safety-critical systems over the years. This results in extensive experience that support the efficacy of the process and techniques recommended by the standards, which are usually the mirror of the state of practice in industry. As a result, standards have shown to be very effective in achieving certification in fields with relatively slow innovation cycles but they exhibit many limitations for emerging technologies where current guidelines are hard to extrapolate [1], [12]. This is the case for dependable autonomous systems, where neither the advanced software algorithms –often based on artificial intelligence– nor the high performance embedded computing features are contemplated by the standards and the proposed techniques are not often applicable. In addition, the high level of connectivity opens the doors to a wide range of security threats that compromise not only privacy but also safety.

These challenges require adapting the traditional certification practices to upcoming technologies. While safety standards define a set of mechanisms to control random hardware faults at runtime, they do not consider such an approach for systematic faults that could arise from the software. In fact, a common assumption of the standards is that systematic faults have to be controlled, tolerated or prevented during the defined development process. However, with increasing software complexity the amount of incidents induced by software glitches is on the rise [2], [3], [5], [10], [11], [14]. In these circumstances, Over-The-Air Software Updates (OTASU), extensively adopted in consumer electronics, provide an efficient and cost-effective method for fixing bugs at operation time, keeping the system up-to-date with the latest security patches or even for adding new functionality. These benefits make OTASU a key technology to stay competitive in many safety-critical markets. The automotive industry is the most prominent example, where the adoption of OTASU for applications such as infotainment, navigation maps or telematic control units is expected to grow exponentially by 2022 [13]. However, the strict certification requirements make most mainstream automotive manufacturers stand aside from updating safety-critical software [5], [9], [13].

This talk will review the main dependability challenges

brought by OTASU to dependable autonomous systems with special focus on safety and security implications over high-performance embedded computing platforms:

- Safety: standards define a clear procedure to manage modifications of safety-critical systems after they have been deployed. However, these modifications often involve a system re-certification, with its associated effort and costs. This approach where an offline impact analysis of each modification is re-assessed by a certification authority, is conceived under the assumption that a safety-critical system will suffer few or even none modifications throughout its lifetime and is not affordable for running frequent software updates. This clashes with the requirements posed by cyber-security standards where critical updates shall be regularly installed. In addition, these challenges are exacerbated by artificial intelligence software as the impact of modification on complex decision making algorithms cannot be exhaustively evaluated beforehand. Similarly, the high-performance computing platforms, often running mixed-criticality software, are subject to intricate dependencies that must be analyzed in conjunction to the final software setup to demonstrate the absence of unacceptable risk. As a result, currently there is not a clear procedure to guarantee that the system remains safe after the update on such complex systems.
- Security: in terms of security, over-the-air updates could be a double-edged sword. On the one hand, OTASU ease the application of security patches, which is an essential cyber-security practice required in standards [7]. On the other hand, it requires access to the network, making the critical systems vulnerable to security threats that could jeopardize safety. Accordingly, extra care should be dedicated to guarantee data integrity and system security in order to be able to preserve system safety.

As a consequence, the adoption of OTASU in dependable autonomous systems requires of novel safety and security co-engineering approaches and mechanisms to achieve certification and gain enough experience and evidence so that they can be adopted in future releases of safety and security standards.

## Acknowledgments

The research presented throughout this paper has received funding from the European Community's Horizon 2020 programme under the UP2DATE project (grant agreement 871465).

## References

[1] I. Agirre, J. Abella, M. Azkarate-Askasua, and F. J. Cazorla. On the tailoring of CAST-32A certification guidance to real COTS multicore architectures. In *2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 1–8, June 2017.

[2] Australian Transport Safety Bureau. In-flight upset event, 240 km north-west of Perth, WA, Boeing Company 777-200, 9M-MRG. https://www.atsb.gov.au/media/24550/aair200503722_001.pdf, March 2007.

[3] BBC News. Tesla recalls 53,000 cars over brake issue. https://www.bbc.com/news/business-39663382, 2017.

[4] EN50128. EN50128:2011 - railway applications: Software for railway control and protection systems, 2011.

[5] Halder, Subir and Ghosal, Amrita and Conti, Mauro. Secure OTA Software Updates in Connected Vehicles: A survey. *arXiv preprint arXiv:1904.00685*, 2019.

[6] IEC. Functional safety of Electrical/Electronic/Programmable Electronic safety-related systems (Second edition), April 2010.

[7] IEC. ISA/IEC 62443 series of standards on industrial automation and control systems security, 2010.

[8] International Organization for Standardization. ISO/DIS 26262 Road Vehicles – Functional Safety, 2009.

[9] John R. Quain. With benefits and risks software updates are coming to the car. https://www.digitaltrends.com/cars/over-the-air-software-updates-cars-pros-cons/, 2018.

[10] Reuters. Volvo Cars recalls 59,000 cars over software fault. https://uk.reuters.com/article/ukvolvocars-recall-idUKKCN0VT0SY, 2016.

[11] Reuters: David Shepardson, Nick Carey. Fiat Chrysler recalls 1.25 million trucks over software error. https://www.reuters.com/article/us-fiatchrysler-recall-idUSKBN1881I6, 2017.

[12] John Rushby. Runtime certification. In Martin Leucker, editor, *Runtime Verification*, pages 21–35, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[13] Michael Smith. Over-the-Air Updates - Is OTA the Future of Cars? https://www.autocreditexpress.com/blog/ota-updates-the-future-of-cars/, 2018.

[14] UK Air investigations. Aircraft Accident Report 4/2007 - Airbus A340-642, G-VATL, 9 February 2005. https://www.gov.uk/aaib-reports/aar-4-2007-airbus-a340-642-g-vatl-9-february-2005, 2007.