The Certification Challenges of Connected and Autonomous Vehicles

Hugues Bonnin

The Certification Challenges of Connected and Autonomous Vehicles

Hugues Bonnin
Continental, Toulouse, France
hugues.bonnin@continental-corporation.com

Abstract— In the context of connected cars, some technologies or approaches are unavoidable: data-centric and big data; AI/ML-based systems; cloud computing and agility. These points are not limited to connected cars, but are trends in all transportation fields at least. In this paper, we analyze the interest of these technologies. Then, we analyze that these techniques are not completely in phase with traditional ways of developing safety critical software because the standards used for this purpose do not rely fundamentally on the same approaches to develop software.

Keywords— autonomous and connected vehicle, safety critical software, certification, big-data, data-centric, cloud computing, Artificial Intelligence/Machine Learning

I. INTRODUCTION

In the beginning of 2017, Continental Automotive founded a new entity: Continental Digital Services France (CDSF). This new entity will propose services around one basic principle: generating added value data from automotive on-board data. Modern vehicles produce a huge quantity of data through their numerous sensors; as they are connected to the internet, this data can be uploaded on off-board infrastructures. Data coming from a large number of vehicles can be mixed together to constitute a very large data-lake. From this data-lake a quasi-infinite number of computations can be imagined to produce added value services based on data: weather, road information, traffic, car maintenance, etc. The data produced by these services may be used in different ways: either it can return to the vehicles, to be used by on-board computers; or it can be used by third party services on the internet.

II. TECHNOLOGICAL TRENDS AND THEIR CHALLENGES

There are some fundamental technologies or approaches that underlie these services.

A. Data Centricity

First of all, these services are data centric, and the volume, the diversity and the velocity of data is why we call it big data. One particular and typical example of data is maps, with their multiple information layers. In the case of connected cars, a lot of use cases produce, enhance or use maps: local weather, road and traffic signs recognition, etc. But maps are only one example of big data: other use cases are data collected for car maintenance or data collected for driver monitoring. Note that these cases are not specific to the automotive field domain: maps, maintenance data, driver or pilot data are also available for aircraft or rail transportation, for example.

B. Artificial Intelligence and Machine Learning

A lot of these use cases are well supported by machine learning approaches, because these techniques allow us to enhance automation, to address the complexity of the relationship between data, and to address the high volume that they often represent. So, even if the technologies behind artificial intelligence and machine learning (AI/ML) are not so recent, their usage has been boosted by the emergence of these data collections. In the context of the automotive industry, this approach is absolutely essential to paving the way to autonomy: analyzing the environment, which is very rich and complex. Visual, radar or sonar image analyses are the main examples. To build and use a correct and sufficiently precise representation of their environment, the sensors of the Autonomous vehicles (AV) have to analyze a very large collection of data. AI/ML approaches are thus fundamental in achieving this, because the intrinsic complexity of this information makes traditional approaches inapplicable.

C. Cloud Computing / Infrastructure

The “large horizon” perception, which is the information based not only on one vehicle, but on all the vehicles present in the same area, in a time frame that ranges from a few seconds to a few days, is the natural extension of the vehicle’s environment. Then AI/ML approaches occur not only in embedded systems, but in off-board systems as well. The collection from multiple vehicles is off-board, on some type of cloud infrastructure. Whatever the nature of this cloud infrastructure and associated services (private, public, mix, SaaS, IaaS, etc.), the big data nature of the collections makes them unavoidable.

These solutions are not only useful to address the high volume of collected data, but they are essential in dealing with scalability. In fact, scalability is an essential dimension of these new services around collected data. The launch of such services is in no case a “one shot” operation. The scalability is multidimensional: the number of vehicles is one dimension; but the type of data may vary in time, too; the accuracy of data could also vary.

D. Agility

This variability implies adopting a global approach that is also agile. Indeed, agility becomes a fundamental pillar of these services, because all is not predefined at the beginning; the developments have to start before all the solutions are found. Development is launched before all aspects of the design and even the specifications are defined. And we anticipate that it

will continue to evolve during operation, making a continuous and permanent life-cycle from the first inception to operation and continuous evolution. The necessity of relying on Agility and DevOps concepts is (at least) twofold: first, the real needs of users are only known when they use their product; second, the speed at which technology is evolving would prevent traditional (V cycle) approaches from integrating them.

We have seen that the services around connected vehicles are closely linked to four fundamental pillars: big data and data centricity, AI/ML based systems, cloud computing, and agile approaches (embracing DevOps). In fact, these considerations are global trends not only in the automotive industry, but also in other fields like aeronautics or railway.

The question then is how to include these services in safety critical functions. Indeed, these services will be part of critical functions, especially in the context of AV, because all the information necessary to build the environment of the vehicle is fused, wherever it comes from. More than that, the more information the vehicle has, the safer it is: then inclusion of this information concurs to the overall safety. It means that we must be confident in technologies and approaches.

III. Current Software Confidence Means

Traditional ways to be confident in software aspects of critical systems are based on conformance of the software development processes to a standard. The standards are written by a community of experts who capture the state of the art of software development. The best practices, which is the fruit of experience feedback, are those which instill confidence in the software built following these practices. Therefore the standards are written based on these best practices, and give the communities that follow these standards a way to build safety critical software with an acceptable level of malfunction risk. But even if these standards are built to be as independent as possible from technologies, some technological breakthrough may show some limits in regards to existing standards, because these breakthroughs deeply stir the state of the art.

In this article, we mainly reference two standards: DO178 [1] and ISO26262 [3]. The first version of DO178 was issued in 1982, the second, DO178A, in 1985, and the DO178B in 1992. The last revision, DO178C, was released in 2012. Without minimizing the effort that has been done to issue the last version, there was no extensive revisit of the way to develop software. ISO26262 was issued in 2012; the second version will be issued in 2018. Part 6 of this standard addresses software development. This part wasn't modified between the two versions. The fundamental principles for software are roughly the same in DO178 and ISO26262. It is to be noted that one part of the revision of ISO26262 has been separated in a new document, and is addressing the Safety Of The Intended Functions (SOTIF) of systems ([4]); it is still under development. In aeronautics, DO200B [2] defines the way to certify databases used in aeronautical embedded systems. Other standards could have been referenced like IEC61508 (generic for industry), EN50128 (railway) or ED153 (European CNS/ATM), or ISO12207; but for the purpose of this article, which is not a detailed study, [1] and [3] are considered sufficiently representative.

IV. Certifications Challenges Brought by the Four Pillars

We think that the technologies exposed in section II constitute a breakthrough point regarding the existing standards. Indeed, we can observe how big data, AI/ML, cloud computing and agile approaches are taken into account in the existing standards.

A. Data Centricity

The first concept is "data centricity." Traditional approaches are more "algorithm-oriented" than "data-oriented". The practices described in standards rely on how to be confident in writing code that manipulates data, not in describing data itself. In a data-centric approach, we would like to evaluate how we can rely confidently on data, without knowing how data is built. Indeed, the way the data is established can be very complex. For example, look at the way a map is established: if it comes from satellites images, the huge amount of transformations from the first signals in the payloads of the satellites, to the pixels displayed in a map on a screen is so complex, that it is unfeasible to trace and make them conform to a standard. The DO200A [2] attempts to formalize these transformations, but is only applicable to a small part of these transformations: the last part of the long chain. We have to find some ways to make robust data, for example including or using already existing intrinsic properties, like redundancy, internal dependencies, diversity, quality factors, etc. We don't know if standardizing a general way to be confident in data is possible, or if only domain and specialized perspectives are real possibilities. Nevertheless, the data centricity has to be fully taken into account for the future development of critical systems.

B. Artificial Intelligence and Machine Learning

The second technology, AI/ML technologies, relies on the first one (data), but adds one supplementary layer in the engineering, which is not taken into account in existing standards: the learning phase. We know learning processes for humans, but we don't know them for machines. Traditional approaches rely on best practices to write software, not on machine training practices during a learning phase. So we have to invent a way to be confident in the systems which will be built through a learning phase: the choice of data sets, the learning process roles, the characteristics of trained models, model verification, etc. Note that this reflection has to be conducted even in the cases where the learning phase is offline (i.e. before operation of the system).

C. Cloud Computing / Infrastructure

The third dimension is the infrastructure: cloud computing. In fact, cloud computing embeds a lot of incompatibilities with traditional approaches: it is made by a lot of COTS (commercial/component off the shelf), when traditional approaches have difficulties dealing with COTS, because their development processes are black boxes, and the confidence is entirely based on the way the development is done. So the only accepted COTS, in the context of a standard, are those in which development has followed this standard. This is a real hard point to build trust in black boxes. The fact that the workgroup writing the latest version of the DO178 standard (DO178C) can't reach an accepted text is proof of the immense challenge.

Moreover, in the case of cloud computing, more than simple software component COTS, these technologies are totally black box services based on software and hardware packages provided by third party specialized companies. The hardware and software infrastructures are highly complex, while traditional approaches tend towards simplicity and transparency. These infrastructures are adapting permanently and changing continuously, contrary to the stability of traditional approaches. Then the general idea to build confidence despite all these "non-conventional" features would be to consider that these black boxes can only provoke a limited number of types of faults. Indeed, traditional approaches are binary: either the confidence is total in the case where the standards are followed, or there is no confidence at all in the other cases. We think that for some systems, we could have confidence that certain classes of fault can't arise. The elements for that are twofold: first, the system can't provoke everything (by its architecture, and its functionality), and there is only a certain range of results that are possible; second, there are some visible characteristics that give confidence: experience made by the service, or renowned experience of the provider, open source access, etc.

D. Agility

The last point is about organization. Agility principles are not those which have been taken as fundamentals in the existing standards that rely on a strong planning and predictability. Agility requires performing activities before they are completely defined, preferring feedback loops to intangible ways to proceed. Indeed DO178C [1] or ISO26262 [3] have a large part devoted to planning, and even if modification can be made to the plans, the general idea is to follow them during development, because they have been accepted by all stakeholders (including authorities in the case of aeronautics). In fact, agile is "change-centric", and traditional approaches are more "plan-centric"; in the first approach, the normal way is continuous change and even uncertainty and unknown; in the second approach, the change is taken into account, but more as an exception than a normality. For example, in DO178C [1], change and problem management are considered in the same chapter.

Agile and existing standards are opposed on another point: oral and written exchange. The first one promotes oral exchanges between the stakeholders, because it is easier, more efficient, and, in the end, less ambiguous. The standards, on the contrary, have the culture of written record of all activities, and leave no room for activity with no written record. A typical activity is coding. In agile approaches significant confidence and autonomy are given to the coder, because it is considered that doing so will be more efficient, even in terms of bug rates. In standards like DO178C [1] or ISO26262 [3], coding is considered to be a very defect-prone activity, and is therefore strongly limited and "fenced" by strict detailed design and strict coding rules.

So we can really ask ourselves if agile approaches do not have something new to bring to traditional approaches. Some experience (see [5]) have shown that agility is compatible with DO178 [1]. But, by definition, agile is compatible with everything; in [5], the general agile principles are applied to the standard, making the standard's activities an objective of the agile production. But doing so doesn't explore the principles and the reasons of its efficiency. It's like doing sport with a Citroën 2CV: you can do it, by adding a motor and reinforcing the chassis; but fundamentally, you don't have the best candidate for sport if you don't completely revisit the design. So making approaches compatible doesn't mean that you use them in an optimal way. The idea here would be more to "rebase" the standards on agile principles.

V. CONCLUSION

We've seen that some technologies or approaches in the context of connected cars are unavoidable: data-centric and big data; AI/ML-based systems; cloud computing and agility. These points are not limited to connected cars, but are trends in all transportation fields at least. Then we analyzed that these techniques are not completely in phase with traditional ways of developing safety critical software because the standards used for this purpose do not rely fundamentally on the same approaches to develop software. The reason is that these standards rely on principles which have not been fundamentally revisited for more than 20 years.

The idea of this paper is neither to provide a magic solution nor to declare that all the standards content is not adequate, but simply to ask the question: are we sure that the lessons learned during the last 20 years of software development have all been explored, regarding the development of safety critical software, to build them more adequately, more efficiently, and, why not, safer? It could be that [4] will address some of these lessons learned; perhaps the principles of Overarching Properties developed by FAA and RESSAC project ([6]) could cover some of these innovative approaches.

This paper focuses on software aspects of confidence in safety critical digital systems. A lot of aspects are linked to this point, but are not developed here: safety aspects could be linked to cybersecurity, and to privacy; the connection between cloud and embedded software is significant (bandwidth, latency, integrity, etc.). Otherwise, the safety process of a digital system is not limited to software aspects: system view or architecture are for example fundamental dimensions to consider; only the global view makes the actual safety of the system. It could be that the technological trends treated in this paper have an impact on all of these related aspects; therefore, the task is far larger than described in the paper…

REFERENCES

[1] DO-178C/ED-12C, Software Considerations in Airborne Systems and Equipment Certification, published by RTCA and EUROCAE, 2012.

[2] DO-200B/ED-76, Standards for processing aeronautical data, published by RTCA and EUROCAE, 2015.

[3] ISO 26262, Road Vehicles - Functional Safety, published by ISO, 2011.

[4] ISO/PAS 21434, Road vehicles - Safety of the Intended Functionality (SOTIF), not yet released, working document of ISO TC22/SC32/WG8.

[5] Agile & Lean development for avionic software, Emmanuel Chenu, Thales Avionics, in Certification Together International Conference, 2011.

[6] An alternative approach to DO178B, Duncan Brown, Rolls Royce, in High Integrity Software conference, 2017.