Adaptation to the Unforeseen: Can We Trust Autonomous and Adaptive Systems?
(Position Paper)

Emil Vassev and Mike Hinchey
Lero – The Irish Software Research Centre, University of Limerick, Limerick, Ireland

Keywords: Smart Vehicles, Autonomous Systems, Adaptive Systems, Adaptability.

Abstract: Autonomous and adaptive systems perform tasks without human intervention and are among the most challenging topics in technology today. Autonomous cars have already appeared on our streets and unfortunately due to some severe accidents they appear to be not as secure as we had hoped them to be. This paper tackles the question of how far we can push the boundary towards achieving such behavior and still provide autonomic operations at least in a certain context with highest safety guarantees to establish trust in autonomous systems.

1 INTRODUCTION

Autonomous systems are becoming ubiquitous and will definitely make it into our daily lives. For example, autonomous cars are already seen on our streets and the first severe accidents prove that they are not as secure as we had hoped them to be. In a perfect world, autonomous cars would be able to share the road with other cars, motorists and bicyclists without any accident. But obviously we are not there yet. The hope, however, is that robotic cars, which are great at monitoring other cars, will be able to identify and avoid any other obstacles on the road.

This position paper presents the authors' vision regarding trust in autonomous cars and adaptive systems in general.

2 ADAPTABILITY - A DESIRABLE PROPERTY?

Without much doubt, the term "adaptive" identifies one of the most challenging topics we currently explore in technology. It identifies systems with the property of being able to react to all situations occurring during its lifetime, both correctly and reliably and the question arises as to whether such a behavior is feasible, implementable, or even desirable.

2.1 Adaptability

Adaptability is conceptually a product of *awareness* and *automation*, or automation through awareness (Vassev and Hinchey, 2012). Awareness is about representing and processing knowledge along with monitoring, while automation is machine replication of human operations.

While automation is more or less about performing a sequence of operations under well-defined conditions, awareness is the capability that drives the automation, i.e., it identifies the specific automation conditions.

Awareness incorporates means by which a computerized machine can perceive events and gather data about its external and internal worlds. Therefore, to exhibit awareness, intelligent systems must sense and analyze components as well as the environment in which they operate.

Determining the state of each component and its status relative to performance standards, or service-level objectives, is therefore vital for an aware system. Such systems should be able to notice changes, understand their implications, and apply both pattern analysis and pattern recognition to determine normal and abnormal states. Generally

Vassev, E. and Hinchey, M.
Adaptation to the Unforeseen: Can We Trust Autonomous and Adaptive Systems? - (Position Paper).
DOI: 10.5220/0006390703660372
In *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2017)*, pages 366-372
ISBN: 978-989-758-242-4

speaking, a self-adaptation should be automation performed to fix abnormal states.

2.2 Need and Feasibility

Without any doubt, we believe that adaptability is a desirable property, but its value really depends on the system context. No adaptability, partial adaptability, full adaptability, and controlled adaptability are different levels or adaptability that will make it both a desirable and realistic property for most of the systems today.

Also, when reasoning on the feasibility of this property, it is important to note that it has both learning and implementation curves, which along with the technology limitations also depend on the thrust in technology. Nowadays, adaptability is a desirable property wherever human control is not possible (e.g., deep space exploration) or a quick reaction is required. For example, to prevent disaster a fuel pump can be automatically stopped in the event of leakage in a fuel system. Other examples where adaptability is not just desired, but even essential, could be related to deep space exploration or any case where human control is not possible due to hazards threatening human lives or due to inability to provide such control.

2.3 Adaptability and Evolution

Our computers are commonly considered as being the most adaptive systems mankind has ever invented and, observing how computers penetrate all aspects of our lives and take control in almost all applications, we have to acknowledge at least the universality of computing equipment. Obviously, the underlying reason for being adaptive and *universal* originates from its very simple basic mechanism to manipulate numbers in the dual system and the ongoing minimization technologies for electronic circuits. However, more important for becoming adaptive, of course, is the programmability of such machines using human intelligence and creativity, and the human's abilities to master complexity using mathematics and computer science technologies.

Adaptability through evolution could be the next level of adaptation where a system evolves to go beyond its original meaning. Such an evolutionary adaptation could tackle, for example, safety properties which are no longer required, because the related hazards had been permanently removed. Another example, could be evolution in the awareness capabilities of the system.

Let's assume that an AI system is programmed to self-adapt its hazard identification capabilities to improve the same or to identify new hazards that were not originally planned to be tackled. Well, we will most probably get to the next level of AI where it evolves and goes beyond its original meaning. This situation can be addressed as technological singularity (Vinge, 1993). Note that the term singularity has been used in math to describe an asymptote-like situation where normal rules no longer apply. For example, an originally programmed AI can stop detecting specific hazards, just because its evaluation criteria has evolved, and these hazards are not hazards anymore. From this point forward, we will be not that far from the moment in the future where our technology's intelligence exceeds our own.

For example, an email spam filter can be loaded with intelligence about how to figure out what is spam and what is not and it will start to learn and tailor its intelligence to us as it gets experience with our particular preferences and habits. However, we may often delete emails that we want to read but do not want to keep in our email box. This can be misinterpreted by the spam filter and it can start filtering these important messages for us.

3 MACHINE LEARNING AND ADAPTABILITY

The term *machine learning* (ML) is provocative as it imputes that machines can learn similar to human beings. In this section we will try to present our interpretation of ML and how it is connected to adaptability.

In general, ML is concerned with computer programs that automatically improve with experience (Mitchell, 1997). To do so, ML tries to use a successful past solution and adapts it to a similar problem, i.e., a sort of reasoning by analogy. This is a principle that can be found in *case-based reasoning* (CBR) systems applied to both simple and structured knowledge representation (Aamod and Plaza, 1994). Such a CBR system works along a cyclic process, by retrieving the most similar cases, reusing the cases in the attempt to solve the problem, revising the proposed solution, if necessary, and retaining the new solution as part of a new case.

ML is perhaps the most advanced field currently explored for self-adaptation in software intensive systems. The problem of adaptation through learning has been a core research issue for a long time. An example of ML connected to adaptability is ML in *dynamically changing environments* where the adaptation is performed through learning and identification of context changes.

But how does ML distinguish from human learning?

Currently, ML is only possible at a small scale where self-adaptation is usually related to formalisms used in Fuzzy Systems, Artificial Neural Networks, and Evolutionary Computation (Baier and Katoen, 2008). ML covers both *symbolic methods* (decision trees and rules, etc.), *sub-symbolic methods* (neural networks, Bayesian networks, etc.) (Berger, 1985) and has several connections with traditional statistics (*discriminant analysis*, *regression analysis*, *cluster analysis*, etc.).

Still, mainly due to its mathematical underpinning, ML is not biased, i.e., humans can be biased in their decisions, but ML is strictly mathematical. Human learning involves different methods and different sources - their “knowledge base” has common facts and strictly personal facts. Experience can be transferred.

4 TRUST IN AUTONOMOUS SYSTEMS

In today’s technologies, the term autonomous plays a major role. It denotes systems which perform their tasks without human intervention as e.g., automatic lawn mowers, smart home equipment, driverless train systems, or autonomous cars. The most challenging question which comes up when following the life cycle of the term “autonomy“ is the potential to construct a system that behaves and operates similarly to, or even better than, a human being. Hence, it is reasonable to discuss how far we can push the boundary towards such behavior and provide autonomic operations at least in a certain context with highest safety guarantees, and finally establish trust in its innocuous operation.

But, will robots ever be able to fully substitute humans?

Our answer is “probably not”. It really depends a lot on the overall impact, not only technological but also political that should be expected from a *universal autonomous system* that successfully replaces human beings. Autonomous systems that will take over the entire traffic control and transportation—yes, because this will eliminate hazards related to human errors, e.g., fatigue. But it is less likely that we will see robots that will ever replace humans in decision making related to social organization, for example.

The means to establish trust in autonomous systems can be roughly described as a twofold objective based on both boundaries and a technical approach:

1. Establish boundaries or a range of adaptation — certain properties (e.g., safety) should be unavoidably held, and thus unforeseen adaptations that may mitigate such properties, should not be allowed without change (human control) in the established adaptation range.
2. Pursue autonomy in a stepwise manner where autonomy can be gradually introduced: *no autonomy*, *partial autonomy*, *controlled autonomy*, and *full autonomy*. Hence in earlier stages, autonomy should be used in less risky domains.

4.1 Dominant Role of Autonomy

It is hard to imagine a system being constructed by a human which adapts itself to all and especially all unforeseen situations as the term unforeseen describes circumstances the human himself has not foreseen.

If we restrict ourselves to some foreseen unforeseen behaviors which we might be able to handle, we have to consider a problem of completeness. Did we cover the whole set of behaviors or did we omit some of the behaviors? This, of course, raises questions of complexity as the number of such situations might be close to infinity and thus, not foreseeable at all. In order to handle such complexity, we have to restrict the adaptability of our systems to a certain context in which we are able to capture all different behaviors, or which at least enables us to classify and cluster such situations. Home environments with a few sensors only might be such a context as well as autonomous transportation systems, e.g., smart trains.

Some other contexts in which autonomy could play a dominant role are contexts where systems operate in environments that change dynamically, e.g., space, ocean, weather stations, etc. In such cases, it is impossible to identify and predefine all possible behaviors. A solution could be related to the use of *granularity in behavior modeling*, i.e., a self-adaptive system should not handle all the possible behaviors, but categories (or classes, or clusters) of possible behaviors. Then, known behaviors shall be classified and let the learning process cope with these.

The awareness-learning process will be able to identify a class of behaviors close to the requested behavior and then pick up a behavior from this class (a low-level behavior). This could be done in a probabilistic manner, when the low-level behavior identification is not possible. Moreover, behaviors based on combinations of predefined behaviors can be identified by a reasoner on an input-output basis.

Therefore, scientists vote again for more math and formalisms in their development but obviously this is much more difficult than it was to prove a non-autonomous system correct. Obviously, it is not just a matter of *logic* and *logical proofs* but it has to incorporate statistical evidence, too, and last but not least, it has to integrate the physical properties of such systems, as e.g., acceleration, loss of weight or the compression of gas under pressure in order to prepare for adaptability. We assume that in order to capture autonomicity in a safe and reliable way, we will see in the near future a convergence of modeling and development techniques based on logics, statistics, and numerics.

4.2 Autonomy and Adaptation Cannot Be 100% Safe

In regards to autonomy and adaptation, the application of formal methods can only add to safety. Even if we assume that proper testing can capture all the autonomy flaws that we may capture with formal verification, with proper use of formal methods we can always improve the quality of requirements and eventually derive more efficient test cases, and consecutively, a safer autonomy. Moreover, formal methods can be used to create formal specifications, which subsequently can be used for automatic *test-case generation*. Hence, in exchange for the extra work put to formally model the autonomy and adaptability of a system, you get not only the possibility to formally verify and validate these features, but also to more efficiently test their implementation.

It is evident that 100% safe autonomy cannot be guaranteed, but when properly used, formal methods can significantly contribute to this by not replacing, but complementing testing. The quantitative measure of how much safety can be gained in autonomy may be regarded in three aspects:

1. Formal verification and validation allows for early detection of autonomy and adaptation flaws, i.e., before implementation.
2. The high quality of autonomy requirements improves the design and implementation of these requirements.
3. Formally specified autonomy requirements assist in the derivation and generation of efficient test cases.

To be more specific, although it really depends on the complexity of the system in question, our intuition is that these three aspects complement each other and together they may help us build a system that has a considerably higher safe autonomy. This principle can further emphasize a systems' ability to autonomously tackle various hazards.

4.3 Correct Adaptations

How do we prove adaptations to be correct and reliable? And is there a difference between such proofs for foreseen and unforeseen behaviors?

Adaptations are related to non-determinism and thus, their correctness proof, if possible, is a tedious task. *Simulation* is one solution. Another one is *probabilistic guarantees*—probabilistic model checking is a powerful technique for formally verifying quantitative properties of systems that exhibit stochastic behavior (Baier and Katoen, 2008).

Probabilistic behavior may arise, for example, due to failures of unreliable components, dynamic environment, etc. Unforeseen behavior cannot be model-checked. It can be eventually simulated (to some extend) through a random generation of the simulated conditions and verified via testing.

4.4 Are Autonomous Systems More Vulnerable?

Considering their adaptive nature, do we expect autonomous systems to be more vulnerable against malicious attacks?

Our definitive answer to this questions is “Yes”. Moreover, the risk of a successful malicious attack is extremely high if the parameterization of their adaptive autonomy is known to the attackers. For example, attackers can use adaptation to open a hole in the security firewall (e.g., a new port can be opened for additional communication link as part of a self-adaptation behavior) that can be unavoidably used for malicious attacks.

An eventual solution could be encryption in communication and why not in autonomy as well. For example, the autonomy-related knowledge base shall be encrypted. Moreover, identification of the autonomy agents shall be required at any level of interaction.

4.5 Connectivity and Safety of Autonomous Transportations

Communication and connectivity are two significant factors that have impact on safety of autonomous transportation. Autonomous smart vehicles are connected, so they are able to share data with other vehicles on the road or with the infrastructure. In general, both vehicle-to-vehicle (V2V) connectivity and vehicle-to-infrastructure (V2I) connectivity help to reduce road accidents, and thus increase the transportation safety. Ultimately, vehicles and road infrastructure are connected via multiple complementary technologies of connectivity such as Wi-Fi, Bluetooth, GPS, DSRC (dedicated short-range communications), etc. Wireless vehicular communication has the potential to enable a class of safety applications that can prevent collisions and save lives as well as improve traffic congestion and fuel efficiency. The effectiveness of these technologies though is highly dependent on cooperative standards for interoperability. For example, DSRC is particularly useful for V2x communications, because it can support very low-latency, secure transmissions and fast network acquisition. It is also considered to be highly robust in adverse weather conditions (Hill, 2015).

The automotive industry has recognized the importance of connected cars by pairing up with other cars and road infrastructure, smart phones, and other smart devices. Moreover, a *swarm of sensors* is used in vehicle internals such as GPS sensors, air pressure sensors, vehicle speed sensor, steering angle sensors, and fire detection sensors among others. In this swarm of sensors, the various electrical components in a car, known as *Electronic Control Units* (or ECUs), are connected via an internal network. Therefore, a smart vehicle exhibits extreme connectivity involving both the external environment and internal ECUs. One of the central challenges in this connectivity is cyber security. Unauthorized access to the external or internal network of a smart vehicle can easily create safety risks. For example, if hackers manage to gain access to a car's Bluetooth or infotainment system, from there they may be able to take control of safety critical ECUs like its brakes or engine and wreak havoc (Toews, 2016).

A contemporary smart vehicle can have over one hundred ECUs and more than one hundred million lines of code. Verification of a complex source code is a tedious task. Further complication is stemming from the fact that vehicle manufacturers source ECUs from many different suppliers, meaning that no one is in control of, or even familiar with, all of a vehicle's source code. Production of complex source code, in particular, increases the security risks and thus the threat of automotive cyber attacks and safety hazards.

5 SELF-DRIVING CAR EXAMPLE

The example presented here should be regarded with the insight that "100% safe autonomy is not possible", especially when the system in question (e.g., a self-driving car) engages in interaction with a non-deterministic and open-world environment (Wirsing, Holzl, Koch, Mayer, 2015) (see Figure 1). What we should do though, to maximize the safety guarantee that "the car would never injure a pedestrian" is to determine all the critical situations involving the car itself in close proximity to pedestrians.

Then, we shall formalize these situations as system and environment states and formalize self-adaptive behavior, e.g., as self-* objectives (Vassev and Hinchey, 2013, Vassev and Hinchey, 2014) driving the car in such situations (Vassev and Hinchey, 2015).

Figure 1: Self-driving Car Interacts with the Environment.

For example, a situation could be defined as "all the car's systems are in operational condition and the car is passing by a school". To increase safety in this situation, we may formalize a self-adaptive behavior such as "automatically decrease the speed down to 20 mph when getting in close proximity to children or a school".

Further, we need to specify situations involving close proximity to pedestrians (e.g., crossing pedestrians) and car states emphasizing damages or malfunction of the driving system, e.g., flat tires, malfunctioning steering wheel, malfunctioning brakes, etc. For example, we may specify a self-adaptive behavior "automatically turn off the engine when the brake system is malfunctioning and the car is getting in close proximity to pedestrians".

6 HOW TO CONSTRUCT RELIABALE AUTONOMOUS SYSTEMS

How can particular modeling techniques, programming concepts and verification methods help to construct reliable autonomous systems?

6.1 Structured Autonomy

Formal methods need to be used in both analysis and code generation. Autonomy requirements need to be properly handled and then both design and implementation should consider formalization of these requirements.

The autonomy part of the system behavior (or the autonomy) needs to be tackled separately - that's it, any system has its purpose (objectives) it needs to follow and functionality that basically supports its objectives. Autonomy is an extra feature (or layer) that is often desirable and its implementation should be definitely separated from that of the system itself.

Autonomy must be structured, implemented, and eventually verified via proper methods, e.g., ARE (Vassev and Hinchey, 2014) and KnowLang (Vassev and Hinchey, 2015). Obviously, the formalization of well-defined properties (e.g., with proper states expressed via boundaries, data range, outputs, etc.) is a straightforward task. However, it is not that easy to formalize uncertainty, e.g., liveness properties. Although, probabilistic theories such as the classical and quantum theories, help us formalize "degrees of truth" and deal with approximate conclusions rather with exact ones, the verification tools for fuzzy control systems are not efficient due to the huge state-explosion problem.

Note that, testing systems implemented over probabilistic theories, is also not efficient, simply because, statistical evidence for their correct behavior may be not enough. Hence, any property that requires a progressive evaluation (or partial satisfaction, e.g., soft goals) is difficult and often impossible to be formalized for use in formally verified systems. Here, it seems the right answer is simulation that will help gain enough statistical evidence for the autonomy behavior correctness.

6.2 Legal and Warranty Issues

Besides the mentioned technical properties, another, often neglected aspect are public laws and regulations the systems have to be conform with. Adaptation will probably make it more difficult to handle such non-functional requirements and requests strict and probably new methods to prove conformity of adaptations with them. For example, engineers currently argue that the most severe obstacles to drive autonomously on our streets are not of technical but of legal nature and concern warranty and guilt.

How could we cover legal and warranty issues in the development and dissemination phases of systems?

We strongly believe that in order to cope with law and warranty, self-adaptive systems need to be introduced in a *stepwise manner*. Laws and concerns will be changed only if the self-adaptive and autonomous systems prove to be better than human operators.

7 CONCLUSION

If we assume a self-driving vehicle loaded with AI, it as a subject to uncertainty due to potential nondeterministic environment it operates on. Often, this lack of determinism is extended by requirements, business conditions, available technology, and the like. Therefore, if we want to construct reliable autonomous systems (e.g., intelligent vehicles), it is important to capture and plan for uncertainty as part of the system's R&D. Failure to do so may result in systems that are overly rigid for their purpose, an eventuality unsafe in their autonomy and adaptation.

Formal methods can assist in the construction of reliable adaptive systems. To do so, formal methods need to be used in both analysis and code generation. Moreover, autonomy requirements need to be properly handled and then both design and implementation should consider formalization of these requirements. For example, contemporary formal verification techniques can be very helpful in verifying safety properties via the formalization of non-desirable system states along with the formalization of behavior that will never lead the system to these states.

ACKNOWLEDGEMENTS

This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero—the Irish Software Research Centre (www.lero.ie).

REFERENCES

Aamod, A., Plaza, E., 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. In *Artificial Intelligence Communications 7(1994):1*, 39-52.

Baier, C., Katoen, J.P., 2008. *Principles of Model Checking*. MIT Press.

Berger, J.O., 1985. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer-Verlag. 2nd edition.

Hill, K., 2015. What is DSRC for the connected car? In *RCR Wireless News,* url: http://www.rcrwireless.com/20151020/featured/what-is-dsrc-for-the-connected-car-tag6

Mitchell, T. M., 1997. *Machine Learning*, McGraw-Hill Science/Engineering/Math. 1st edition.

Toews, R., 2016. The Biggest Threat Facing Connected Autonomous Vehicles is Cybersecurity. In *TC Sessions*. https://techcrunch.com/2016/08/25/the-biggest-threat-facing-connected-autonomous-vehicles-is-cybersecurity/

Vassev, E., Hinchey, M., 2012. Awareness in Software-Intensive Systems. In *IEEE Computer 45(12)*, 84–87.

Vassev, E., Hinchey, M., 2013. Autonomy requirements engineering. In *IEEE Computer 46(8)*, 82–84.

Vassev, E., Hinchey, M., 2014. Autonomy Requirements Engineering for Space Missions. In *NASA Monographs in Systems and Software Engineering*. Springer.

Vassev, E., Hinchey, M., 2015. Knowledge representation for adaptive and self-aware systems. In *Software Engineering for Collective Autonomic Systems*, Volume 8998 of LNCS. Springer.

Vinge, V., 1993. The coming technological singularity: How to survive in the post-human era. https://www-rohan.sdsu.edu/faculty/vinge/misc/singularity.html

Wirsing, M., Holzl, M., Koch, N., Mayer, P. (eds.), 2015. *Software Engineering for Collective Autonomic Systems*. Volume 8998 of LNCS. Springer.