DOI: 10.15439/2022F273
ISSN 2300-5963 ACSIS, Vol. 31

Rethinking Safety in Autonomous Ecosystems

David Halasz and Barbora Buhnova
Faculty of Informatics, Masaryk University
Brno, Czech Republic
{halasz, buhnova}@mail.muni.cz

***Abstract*—As autonomous cyber-physical systems are responding to the dynamism of our hyper-connected digital world, they are forming so called dynamic autonomous ecosystems, which require a change in methods ensuring their safe behavior. Within this change, reactions to predictable scenarios need to be replaced with adaptability to the unpredictable context, with gradual safety mechanisms, able to decide whether or not to trigger a certain mitigation procedure. In this paper, we outline our vision towards evolution of safety mechanisms to support dynamic and self-adaptive architectures of autonomous ecosystems. We are proposing an approach to address this research problem with the help of trust and reputation combined with gradual adaptation of safety procedures at runtime.**

I. Introduction

THE growing demand for complex autonomous cyber-physical systems is stimulating their advancement in the direction of forming cooperative and collaborative autonomous ecosystems [1]. Such autonomous ecosystems, i.e. dynamic autonomous systems of systems, can provide a higher degree of autonomy and are capable of adapting to previously unknown situations. At the same time, however, their dynamic and self-adaptive nature is making it very challenging to ensure their safe and secure behavior, both at the individual level as well as at the level of the ecosystem as a whole [2].

The recent rapid development in autonomous driving is indicating that new autonomous systems might be joining city ecosystems sooner than the cities need to get ready to ensure the safety of these ecosystems as a whole, which is challenging not only technically but also from the perspective of understanding the ways in which societies perceive safety and trust in these autonomous systems.

Even though there has been substantial progress in the research of the methods ensuring safety in individual autonomous systems, the methods are falling short on the larger scale of autonomous ecosystems, in which the individual autonomous systems dynamically join and leave the ecosystem and interact with each other in a decentralized manner [1]. In this environment where multiple autonomous systems operate in the same physical space with high level of complexity and dynamic context changes, existing safety-assurance methods on the level of each individual systems might become too rigid to support the overall ecosystems.

Borrowing from the ways in which our societies ensure safety of its members, one of the most promising ways towards the safety of dynamic autonomous ecosystems is through the mechanisms of adaptive safety reflecting the actual safety risks in a given situation, which can be understood based on the trust and trustworthiness of the ecosystem members one interacts with [3], [2].

To stimulate the progress in this emerging field, the aim of this paper is to examine the problem of safety-assurance in dynamic autonomous ecosystems and envision an approach for adaptive safety in the ecosystems. Namely, we set the foundations for a new approach to adaptive safety, responding to the level of trust among autonomous systems. To this end, we first identify and present five scenarios of the key challenges related to safety in dynamic autonomous ecosystems, and then propose a framework to support adaptive safety in the ecosystems.

The structure of the paper as follows: Section 2 identifies the key challenges of adaptive safety in dynamic autonomous ecosystems and presents the example scenarios. Section 3 discusses research related to the addressed problem, followed with Section 4 presenting a solution and proposing a framework to support adaptive safety in dynamic autonomous ecosystems. Section 5 discusses assumptions and limitations made in designing the approach, which is followed with a conclusion and summary of future work.

II. Problem Description

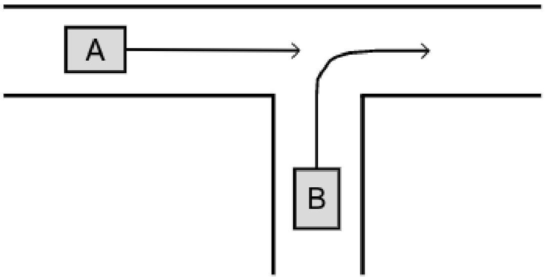
Safety as it is perceived and enforced on the level of individual autonomous cyber-physical systems is falling short on the magnitude of ecosystems [2]. The techniques that are capable of keeping a single system safe are not scaling to a dynamic ecosystem where member systems are heterogeneous and can join or leave the ecosystem at any time. This context requires an adaptive approach to safety that should be based on a kind of classification among member systems. One of the most promising strategies that only started to emerge recently, is to adapt safety to the level of trust among components within the whole ecosystem. In that context, a component of the autonomous ecosystem that is reported as untrusted by other ecosystem members (impacting its reputation within the ecosystem) might fall under (temporary) safety supervision and control, safeguarding its trustworthy operation.

To set the context for this research, this section identifies and discusses five challenges (described in the individual subsections) that need to be resolved to set the foundation of our approach to trust-driven adaptive safety in dynamic autonomous ecosystems. Our aim is to provide a solution to these challenges and propose a safety assurance framework for autonomous ecosystems on top of it.

A. Intentional vs. unintentional behavior

When an action done by an autonomous system has been classified as malicious, knowledge about the intent of this system can be an important decision factor when selecting the right kind of reaction. Unintentional malicious behavior can happen due to a malfunction in one of the components of the autonomous systems, delay in network communication or a software bug. On the other hand, it is possible to design a system that behaves in a harmful way in certain situations and tries to inflict as much damage as possible [2].

Fig. 1. Example scenario where intent classification is important



Consider an example in Figure 1 where two autonomous vehicles are meeting at an intersection. If vehicle B sends false information about its speed to vehicle A, relying on this information would cause a crash. In case the sensor responsible for measuring the speed of vehicle B is malfunctioning, a good course of action for vehicle A is to slow down and let vehicle B merge into the lane without interfering with it. However, if vehicle B is programmed to crash into vehicle A, the mitigation would not be sufficient in avoiding a crash.

B. Supervision awareness

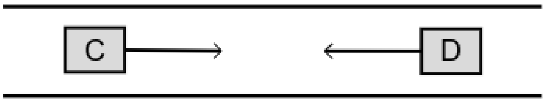
Unintentional malicious behavior can be sometimes corrected by letting the system know that it is behaving the wrong way. However, in some cases this operation would equip the system with additional information that could leveraged against another systems or to compromise the integrity of the ecosystem as a whole [2].

In the past decade, there were multiple scandals where vehicle manufacturers installed supervision awareness detection in their products [4]. The goal of this piece of software was to detect whether the vehicle is under emissions test or used by its owner. Based on the detected context, the ECU was instructed to reduce the CO_2 emissions by lowering the overall torque and power produced by the engine. This example can be simply extended to the domain to the autonomous ecosystems, where a member system can behave differently if it is being monitored and start behaving maliciously when it detects that the supervision is suspended.

C. Misclassification of behavior

When deciding whether an action of a system is malicious or not, there is always a margin of error. No classification technique can be always perfect and this inherently carries some danger when using such technique to make decision about enforcing safety mechanisms. If a behavior is incorrectly classified as malicious, reactions to this *false-negative* scenario can unnecessarily limit the functionality of the system. Furthermore, the result can influence any future interaction with this system can be restricted in the ecosystem. On the other hand, when a malicious action is wrongly classified as regular or safe behavior, this case is a *false-positive* and it can allow the system to cause even more damage then it originally intended to inflict on the ecosystem [2]. Some kind of compensation between these two extremes is necessary to both maintain the functionality and also ensure safety.

Fig. 2. Example scenario where misclassification can cause issues



Consider a scenario in Figure 2 where both misclassification cases can cause issues. If autonomous vehicle C is not capable of detecting the malicious intent of autonomous vehicle D, the situation can escalate into a frontal crash. Meanwhile, if vehicle D is wrongly classified as malicious, the triggered collision avoidance mechanism can slow down vehicle C more than it would be necessary with a correct classification. This would slow down the whole intersection for a longer amount of time that could affect other autonomous vehicles as well.

D. Feedback loops

The possibility of repetitive misclassification in multiple systems that interact with each other can create an even more challenging issue. Safety mechanisms invoked by one system can be interpreted by other systems as malicious. Any reaction to this false-negative can be also interpreted as malicious which can cause a gradual triggering of more and more strict safety features in every interacting system. This can even lead to a permanent stall of the whole ecosystem, especially if one of the member systems has been intentionally designed to cause an issue like this. This possibility should be considered when designing the safety architecture of an ecosystem [2].

Fig. 3. Example scenario where feedback loop can cause dangerous behavior of both vehicles

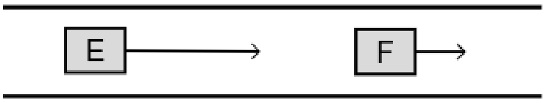
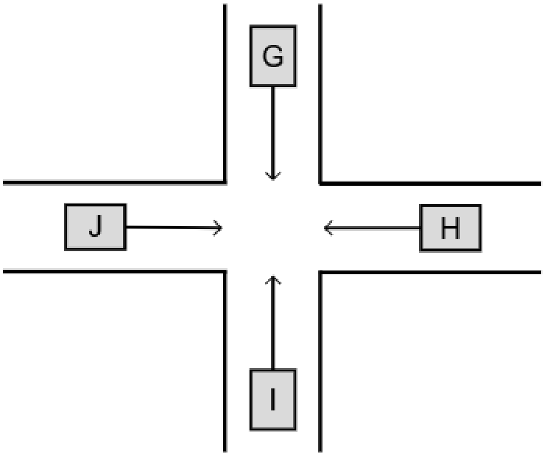


Figure 3 shows two autonomous vehicles heading in the same direction, where vehicle E has a higher speed rating and eventually it would overtake vehicle F. If vehicle F does not recognize the overtaking action and classifies the acceleration of vehicle E as malicious, it can accelerate to a higher speed to avoid a possible collision. This behavior can be interpreted by vehicle E as malicious and as a reaction it could decrease its speed. As there is no reason for vehicle F to increase its speed anymore, it can adjust its speed to the initial one. If vehicle E returns to its original speed, the whole situation can repeat itself from the beginning. The other possible outcome is to adjust the speed of both vehicles to the same speed, which would be not optimal for vehicle E as it is capable of higher speeds for a longer amount of time.

E. Compatibility

In any autonomous ecosystems there is a possibility of having heterogeneous member systems, manufactured by various vendors, using different implementations that not necessarily provide the same (safety) features. In order to ensure the safe behavior of the ecosystem, it is necessary to be able to provide some kind of backwards compatibility for systems with a reduced set of safety features. Alternatively, the ecosystem should be able to (at least temporarily) equip these systems with some kind of common safety mechanism. An extreme edge case of this problem is when a human is interacting with the ecosystem, which can be interpreted as a member system with zero compatibility and no possibility to receive a new safety mechanism.

Fig. 4. Example scenario for a critical compatibility issue



The example in Figure 4 shows four vehicles meeting at an intersection. Autonomous vehicles G and H use the most modern safety assurance framework which provides a solution for all the problems stated in Section 2. Autonomous vehicle I uses a different implementation in which some of these problems are not fully covered. Lastly, vehicle J is driven by a human who has no or minimal knowledge about what kind of software is running on the three autonomous vehicles.

III. State of the Art and Related Work

Safety can be interpreted differently in each domain. Our research found that the most relevant definitions for our purposes are "the ability of a distributed application and its parts to continue operating in a safe manner during and after a transformation" [5] and the "avoidance of hazards to the physical environment" [6].

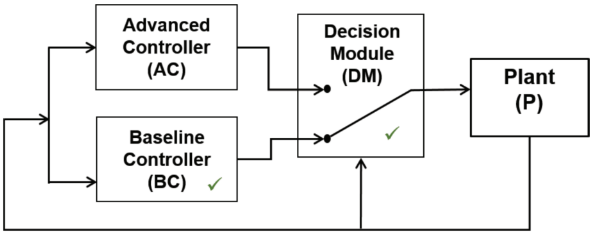
A. Safety in Autonomous Vehicles

Research in the area already covers most of the safety aspects of individual autonomous systems. Collision avoidance [7], communication security and recovering from attacks [8], [9] in the subdomain of autonomous vehicles are not dealing with safety on the magnitude of an ecosystem as a whole. Safety assurance in vehicle platooning [10] on the other hand is close to the area of interest, however, it does not provide answers to all the problems stated in Section 2.

B. Simplex architecture

The concept of "using simplicity to control complexity" [11] implemented by the Simplex architecture is an interesting approach to ensuring safe behavior of a system, popular in control systems and beyond. The core of the idea is to split up a system into a complex component (advanced controller) supporting all its ordinary behavior and a simpler component (baseline controller) that is only intended to resolve critical situations. A decision module between these two components can select which one should be enabled in certain situations [12]. While combining multiple simplexes can be a viable solution in having a complex system of systems where each system is responsible for its own safe behavior [13], [14], they are not designed to deal with uncertain situations. Also they can be prone to feedback loops and the lack of the granularity of the safety assurance can cause problems if a misclassification occurs [2].

Fig. 5. The simplex architecture [14]

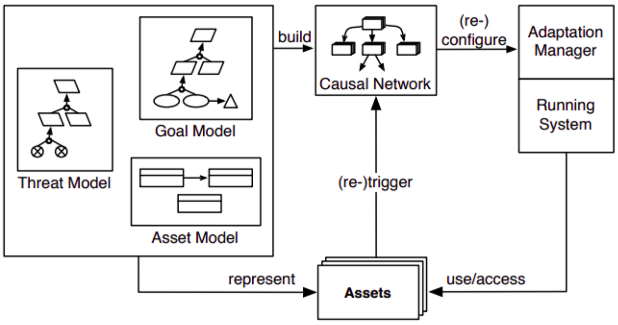


C. Self-adaptation

Self-adaptive cyber-physical systems are capable of handling uncertain situations [15]. This adaptability can be achieved by techniques such as runtime model querying [16] or Monitor Analyze Plan Execute with Knowledge (MAPE-K) [17] feedback loops. *Security* that is defined as "something "concerned with protecting assets from harm" can be also enforced in an adaptive way that is being evaluated during runtime [18]. This and techniques like Adaptive Control Lyapunov Functions aCLFs [19] can be also applied to (instead of security) enforce the *safety* of an autonomous system.

Although these solutions are well designed for autonomous systems, they do not provide answers to securing an ecosystem as a whole. Scaling a feedback loop by distributing it to multiple systems face difficulties as member systems are not always collaborative.

Fig. 6. Framework to support adaptive security [18]



D. Safety in distributed ecosystems

When considering safety in largely distributed ecosystems, wireless networks become a noteworthy source of knowledge, even though their most important aspect is communication security [20], [21]. The way how ad-hoc and self organizing mesh networks [22], [23] work strongly resembles the dynamicity in autonomous ecosystems. Techniques from this subdomain [24], [25], [26] might be useful in our context. It is however limiting that they can only cover the cyber part of a cyber-physical ecosystem. Cutting off a node from a network can surely increase safety, however, ignoring a robot or an autonomous vehicle in a similar way can cause more problems than it solves. Moreover, such scenario could hint a malicious system about no longer being monitored, which introduces the problem of supervision awareness in the ecosystem.

E. Classification of behavior

Determining if an action done by a system is malicious or not is the most important factor when selecting the appropriate reaction in other member systems. Due to the dynamicity of the ecosystem, such classification has to be conducted continuously and at real-time. In an ideal case, each member system could have its own model constructed [27] and propagated that could be queried by other systems to decide on further actions. The approach is called *models@run.time* [16] and it is intended to be used in scenarios that were not taken into account when the system had been designed [28]. The problem with this approach is the requirement of a valid model for each member system, which is not always possible to construct. Another issue is the distribution of these models and the fact that sharing them with malicious systems might equip them with knowledge about vulnerabilities and increase the overall attack surface [2].

IV. Proposed solution

Ensuring safe and secure behavior of autonomous cyber-physical ecosystems is a challenging task and it requires a new approach on how relationships between individual entities are perceived. The inherent complexity and the dynamic context changes of the consistency of such ecosystems cannot be solved during design time. Therefore, any proposed solution has to able to handle previously unexpected or uncertain situations during runtime. Referring to our previous work [2] and Liu et al. [29], we believe that leveraging real-time evaluated trust among ecosystem components can provide sufficient input to make real-time decisions about safety in dynamic autonomous ecosystems [30].

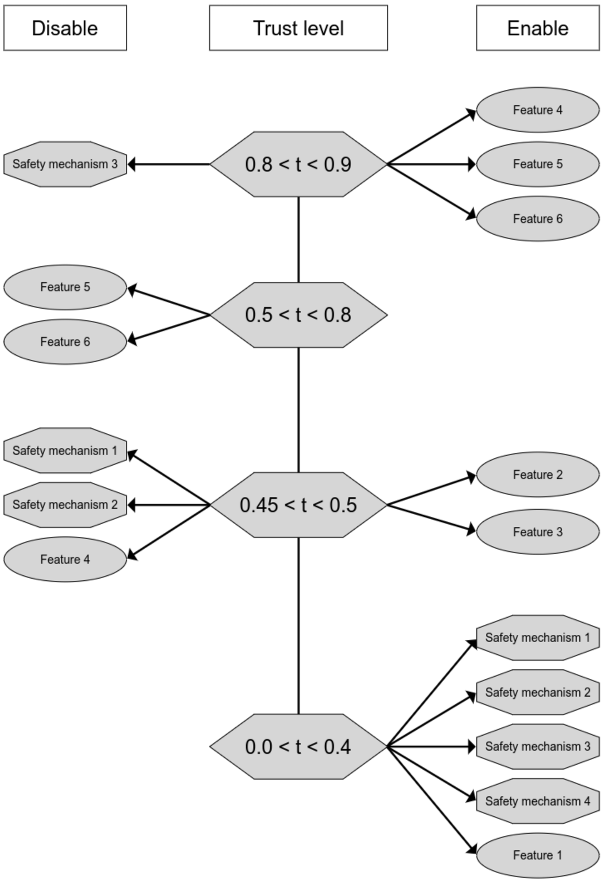
The definition of trust can to be borrowed from different branches of science [31], such as Psychology [32], Philosophy [33] and Organizational Management [34]. Simply put, autonomous systems shall understand trust similarly as we humans do.

Most of the research in the area is conducted around the qualitative understanding of trust [31], i.e. classifying it into a binary form to either trust or not to trust. These approaches, however, due to the lack of the granularity of their output, are prone to misclassification. Since the appearance of the Internet of Things, there are some promising approaches that are able to assess trust quantitatively [29], e.g. into a percentage. Due to the higher variety in the output, such methods are statistically less likely to be far away from the right result in case of an error in the trust calculation. It is important to mention, that trust can be calculated directly (trust) from a target system or obtained indirectly (reputation) from other systems that have had former interactions with the target system [35], [36], [37]. In some cases the two can be merged into a combined value with predefined or dynamic weights.

Any input consumed by our solution has to be more granular than a binary, e.g. *to trust* or *not to trust*. This should significantly reduce the chance of errors happening due to misclassification as the distance between the ideal and the actual output is statistically lower than in a binary situation. This granularity should be mirrored in the safety enforcement with a graduality of triggering safety mechanisms or exposing features towards other autonomous systems [2].

The decision tree for a single autonomous system of an example safety mechanism is shown in Figure 7. For simplification, it assumes a numeric input coming from a trust model with a decimal number between 0 and 1. This trust level is being continuously recalculated in real-time against any interacting autonomous system and used to decide which features should be exposed or concealed and which safety mechanisms should be triggered in individual situations. A low trust level allows a minimal set of features and a large number of safety mechanisms and as the trust level is growing, the trend is gradually reversing. It is important, that safety mechanisms are also available on the highest level of trust and the system can move to a lower trust level at any time during its operation.

Fig. 7. Example decision tree: actions to take based on trust [2]



A. Safety Assurance Framework

The envisioned framework supporting trust-based adaptive safety is drafted in Figure 8. Its core components are the Trust Model, the Decision Tree and the Safety Module connecting these two. 1) The *Trust Model* calculates a *Trust Value* of a target system based on inputs from sensors and a reputations propagated by other actors of the ecosystem. 2) The *Trust Value* calculated by the model is consumed by the Safety Module and propagated to other systems doing similar calculations. 3) The *Safety Module* using the Decision Tree selects what safety mechanisms should be enabled or disabled and what features can be exposed to or concealed from the target system. 4) The *Decision Tree* allows the possibility to alter itself either by a software update or by the system itself using a self-adaptive technique.

The process of trust calculation is continuously triggered by the *Safety Module* after each adaptation cycle. This ensures that the system has the most recent information about how much a target system can be trusted at all time.

In case the *Trust Value* has been assessed wrongly, granularity of the trust output combined with the gradual triggering of safety features significantly decreases the margin of error and its consequences. In case a system with high level of trust behaves maliciously, the fast recalculation of the *Trust Value* can not just help the attacked system to quickly adapt, but this new information can be propagated to other members of the ecosystem. The spreading reputation can influence trust computations in other systems, ensuring their safety features are prepared for a future encounter with a malicious system. Furthermore, fresh reputation information might provide means to end a feedback loop introduced by wrong trust calculations.

The real-time recalculation and quick reaction time of triggering safety features should address most of the possible issues related to the detection of intent. Due to the continuous recalculation of the *Trust Value*, supervision awareness is also addressed by this technique. A malicious system that previously maximized its trustworthiness to access a certain feature can be quickly detected. We envision that trust propagation in ecosystems would create clusters of safely operating member systems and push malicious ones to the periphery. Interaction with them should not be completely severed as knowledge about them can be helpful in preventing further harm in the future.

If trust is calculated by using predictive simulations via Digital Twins [3], the same Digital Twins can be also used to partially determine the capabilities of other systems. This equips the Safety Assurance Framework with critical information regarding compatibility and might prevent certain cases of feedback loops. In this case the framework has to be able to receive Digital Twins and run predictive simulations even independently from its *Trust Model*. It is also necessary to consider situations when digital twins are not available or predictive simulation is not an option, e.g. in case of a human. Most importantly, the *Decision Tree* has to be constructed in a way that it handles these situations.

B. Example scenario

Consider a scenario of two autonomous vehicles in Figure 9 leveraging our framework. Both vehicles can in advance assess how much they trust each other. In case if that information is available, they can rely on reputation propagated by other actors of the ecosystem as well. From the perspective of vehicle A, when trust towards the malicious vehicle B is calculated as 0.7, the system would initially trigger safety mechanisms only for avoiding a frontal collision by moving to the right side of the road. As the *Trust Value* is high enough, this information would be communicated to vehicle B. If the malicious behavior is becoming more obvious, the constantly recalculated *Trust Value* first drops to 0.4, vehicle A begins to reduce its speed and tries to find a course that would minimize the risk of a collision. As the vehicles are getting closer to each other and the *Trust Value* reaches 0.2, an active collision avoidance mechanism takes over the control and tries to keep safe distance from vehicle B.

In a reversed scenario if the wrongly calculated *Trust Value* is 0.2, the same active collision avoidance mechanism is controlling vehicle A. Vehicle B detects this behavior and

Fig. 8. Framework to support Trust-based Adaptive Safety

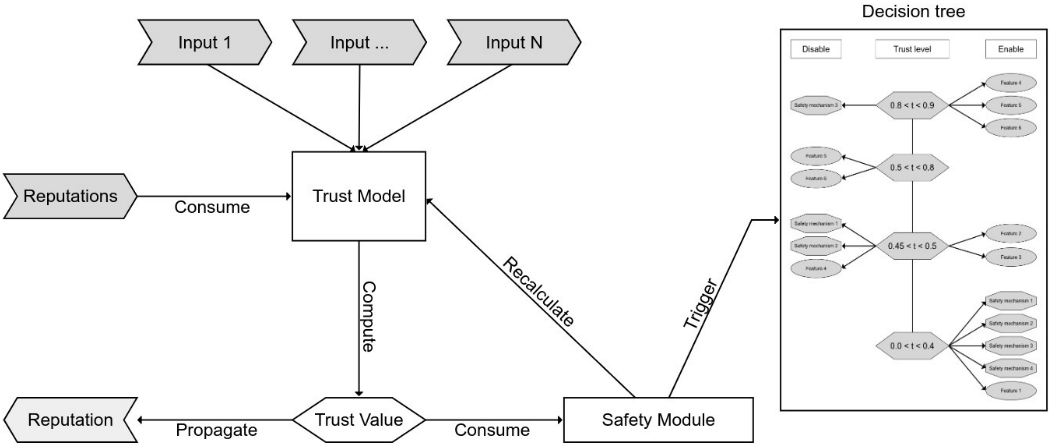
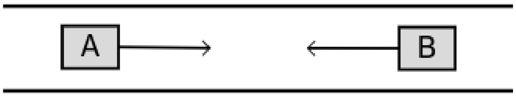


Fig. 9. Example scenario



also starts behaving more cautiously, which increases the *Trust Value* towards it to 0.5. This leads to more information sharing between the two vehicles and the increased number of inputs increases the *Trust Value* to 0.9 by the time the two vehicles meet and they both move to their right side of the road and continue with an increased speed.

After their encounter, both vehicles store the final *Trust Value*, that would prevent them from getting into similar scenarios with each other. In the meantime, all calculated *Trust Values* are propagated into the ecosystem (in terms of a reputation of each individual vehicle), which should reduce any misclassification for other actors.

V. Discussion

Our approach proposes a paradigm shift in comparison with existing solutions. This paper is exploratory in nature and intends to start a community discussion about future steps in this direction.

Even though trust is the designated decision factor in our approach, it is only considered as an input to the proposed mechanism. The *Trust Model* is treated as a black box and its main requirement is to produce a non-binary granular output. Having a safety mechanism decoupled from its input allows us to have additional flexibility. Autonomous systems can implement different *Trust Models* [30] and it might also happen that decision factors other than trust will be consumed by the solution.

Our future plans are to finalize the specification of the Safety Assurance Framework and define its input and output interfaces. Meanwhile, our research team is reviewing trust computation methods that can be consumed by the proposed solution. After both are ready and available, our plan is to reach out to automotive companies and work together with them to validate the framework on real-life case studies.

VI. Conclusion

Due to the rising complexity of autonomous ecosystems, new software-architecture mechanisms are necessary to respond to the dynamicity of changes while ensuring safety even in uncertain situations. In this work, we propose to address this challenge via mechanisms to gradually enable safety mechanisms based on the assessed level of trust towards other members of the ecosystem, in combination to new ways of assessing the trustworthiness of individual system components. Furthermore, we describe the fundamentals of a Safety Assurance Framework that would support this mechanism. In our next steps we plan to create a more thorough design of the framework and validate it on case studies provided by possible partners from the industry. We believe that this idea will evolve into a set of prototypical tools supporting promoting of safe autonomous ecosystems.

Acknowledgment

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

[1] R. Capilla, E. Cioroaica, B. Buhnova, and J. Bosch, “On autonomous dynamic software ecosystems,” *IEEE Transactions on Engineering Management*, pp. 1–15, 2021. doi: 10.1109/TEM.2021.3116873

[2] D. Halasz, “From systems to ecosystems: Rethinking adaptive safety,” in *17th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS ’22)*. IEEE, 2022. doi: 10.1145/3524844.3528067

[3] E. Cioroaica, T. Kuhn, and B. Buhnova, “(do not) trust in ecosystems,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*. IEEE, 2019. doi: 10.1109/ICSE-NIER.2019.00011 pp. 9–12.

[4] M. Contag, G. Li, A. Pawlowski, F. Domke, K. Levchenko, T. Holz, and S. Savage, "How they did it: An analysis of emission defeat devices in modern automobiles," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017. doi: 10.1109/SP.2017.66 pp. 231–250.

[5] P. McKinley, S. Sadjadi, E. Kasten, and B. Cheng, "Composing adaptive software," *Computer*, vol. 37, no. 7, pp. 56–64, 2004. doi: 10.1109/MC.2004.48

[6] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security, and sustainability of mission-critical cyber–physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 283–299, 2012. doi: 10.1109/JPROC.2011.2165689

[7] G. Li, Y. Yang, T. Zhang, X. Qu, D. Cao, B. Cheng, and K. Li, "Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102820, 2021. doi: 10.1016/j.trc.2020.102820. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20307257

[8] S. Bouchelaghem, A. Bouabdallah, and M. Omar, "Autonomous Vehicle Security: Literature Review of Real Attack Experiments," in *The 15th International Conference on Risks and Security of Internet and Systems*, Paris, France, 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03034640

[9] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou, "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles," *Ad Hoc Networks*, vol. 90, p. 101823, 2019. doi: 10.1016/j.adhoc.2018.12.006 Recent advances on security and privacy in Intelligent Transportation Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870518309260

[10] J. Axelsson, "Safety in vehicle platooning: A systematic literature review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 1–13, 08 2016. doi: 10.1109/TITS.2016.2598873

[11] L. Sha, "Using simplicity to control complexity," *IEEE Software*, vol. 18, no. 4, pp. 20–28, 2001. doi: 10.1109/MS.2001.936213

[12] D. Seto, B. Krogh, L. Sha, and A. Chutinan, "The simplex architecture for safe online control system upgrades," in *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*, vol. 6, 1998. doi: 10.1109/ACC.1998.703255 pp. 3504–3508 vol.6.

[13] P. Vivekanandan, G. Garcia, H. Yun, and S. Keshmiri, "A simplex architecture for intelligent and safe unmanned aerial vehicles," in *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2016. doi: 10.1109/RTCSA.2016.17 pp. 69–75.

[14] D. Phan, J. Yang, M. Clark, R. Grosu, J. Schierman, S. Smolka, and S. Stoller, "A component-based simplex architecture for high-assurance cyber-physical systems," in *2017 17th International Conference on Application of Concurrency to System Design (ACSD)*, 2017. doi: 10.1109/ACSD.2017.23 pp. 49–58.

[15] H. Muccini, M. Sharaf, and D. Weyns, "Self-adaptation for cyber-physical systems: A systematic literature review," in *2016 IEEE/ACM 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2016. doi: 10.1145/2897053.2897069 pp. 75–81.

[16] N. Bencomo, R. France, B. Cheng, and U. Aßmann, Eds., *Models@run.time: foundations, applications, and roadmaps*, ser. Lecture notes in computer science. Germany: Springer, Dec. 2014. ISBN 978-3-319-08914-0 Dagstuhl Seminar 11481 on models@run.time held in November/December 2011.

[17] P. Arcaini, E. Riccobene, and P. Scandurra, "Modeling and analyzing mape-k feedback loops for self-adaptation," in *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2015. doi: 10.1109/SEAMS.2015.10 pp. 13–23.

[18] M. Salehie, L. Pasquale, I. Omoronyia, R. Ali, and B. Nuseibeh, "Requirements-driven adaptive security: Protecting variable assets at runtime," in *2012 20th IEEE International Requirements Engineering Conference (RE)*, 2012. doi: 10.1109/RE.2012.6345794 pp. 111–120.

[19] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *2020 American Control Conference (ACC)*, 2020. doi: 10.23919/ACC45564.2020.9147463 pp. 1399–1405.

[20] L. S. Rutledge and L. J. Hoffman, "A survey of issues in computer network security," *Computers & Security*, vol. 5, no. 4, pp. 296–308, 1986. doi: 10.1016/0167-4048(86)90050-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0167404886900507

[21] M. S. Siddiqui, "Security issues in wireless mesh networks," in *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, 2007. doi: 10.1109/MUE.2007.187 pp. 717–722.

[22] A. J. Fehske, I. Viering, J. Voigt, C. Sartori, S. Redana, and G. P. Fettweis, "Small-cell self-organizing wireless networks," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 334–350, 2014. doi: 10.1109/JPROC.2014.2301595

[23] R. Katulski, J. Stefański, J. Sadowski, S. Ambroziak, and B. Miszewska, *Self-Organizing Wireless Monitoring System for Containers*. Springer, 08 2009, pp. 164–172. ISBN 978-3-642-03840-2

[24] S. Desilva and R. Boppana, "Mitigating malicious control packet floods in ad hoc networks," in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 4, 2005. doi: 10.1109/WCNC.2005.1424844 pp. 2112–2117 Vol. 4.

[25] M.-Y. Su, K.-L. Chiang, and W.-C. Liao, "Mitigation of black-hole nodes in mobile ad hoc networks," in *International Symposium on Parallel and Distributed Processing with Applications*, 2010. doi: 10.1109/ISPA.2010.74 pp. 162–167.

[26] A. Naveena and K. R. L. Reddy, "Malicious node prevention and mitigation in manets using a hybrid security model," *Information Security Journal: A Global Perspective*, vol. 27, no. 2, pp. 92–101, 2018. doi: 10.1080/19393555.2017.1415399. [Online]. Available: 10.1080/19393555.2017.1415399

[27] S. Kent, "Model driven engineering," in *Integrated Formal Methods*, M. Butler, L. Petre, and K. Sere, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. ISBN 978-3-540-47884-3 pp. 286–298.

[28] M. Barkowsky, T. Brand, and H. Giese, "Improving adaptive monitoring with incremental runtime model queries," in *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2021. doi: 10.1109/SEAMS51251.2021.00019 pp. 71–77.

[29] L. Liu, M. Loper, Y. Ozkaya, A. Yasar, and E. Yigitoglu, "Machine to machine trust in the iot era," in *Proceedings of the 18th International Conference on Trust in Agent Societies - Volume 1578*, ser. TRUST'16. Aachen, DEU: CEUR-WS.org, 2016, p. 18–29.

[30] D. Iqbal and B. Buhnova, "Model-based approach for building trust in autonomous drones through digital twins," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2022.

[31] J.-H. Cho, K. Chan, and S. Adali, "A survey on trust modeling," *ACM Comput. Surv.*, vol. 48, no. 2, oct 2015. doi: 10.1145/2815595. [Online]. Available: 10.1145/2815595

[32] J. B. Rotter, "Interpersonal trust, trustworthiness, and gullibility." *American psychologist*, vol. 35, no. 1, p. 1, 1980. doi: 10.1037/0003-066X.35.1.1

[33] B. Lahno, "On the emotional character of trust," *Ethical theory and moral practice*, vol. 4, no. 2, pp. 171–189, 2001.

[34] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An integrative model of organizational trust," *The Academy of Management Review*, vol. 20, no. 3, pp. 709–734, 1995. doi: 10.2307/258792. [Online]. Available: http://www.jstor.org/stable/258792

[35] A. I. A. Ahmed, S. H. Ab Hamid, A. Gani, M. K. Khan *et al.*, "Trust and reputation for internet of things: Fundamentals, taxonomy, and open research challenges," *Journal of Network and Computer Applications*, vol. 145, p. 102409, 2019. doi: 10.1016/j.jnca.2019.102409

[36] A. Hbaieb, S. Ayed, and L. Chaari, "A survey of trust management in the internet of vehicles," *Computer Networks*, vol. 203, p. 108558, 2022. doi: 10.1016/j.comnet.2021.108558. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621004758

[37] W. Abdelghani, C. A. Zayani, I. Amous, and F. Sèdes, "Trust management in social internet of things: a survey," in *Conference on e-Business, e-Services and e-Society*. Springer, 2016. doi: 10.1007/978-3-319-45234-0_39 pp. 430–441.