Autonomous Systems, Trust, and Guarantees

Nima TaheriNejad
TU Wien, Vienna, Austria

Axel Jantsch
TU Wien, Vienna, Austria

Andreas Herkersdorf
Technical University of Munich, Munich, Germany

Editor's notes:
Trustworthiness is key for the acceptance of autonomous systems. The authors advocate deterministic methods with hard-bounded corridors for operational parameters to guarantee dependable autonomy considering both functional and extra-functional properties.

—Selma Saidi, TU Dortmund

In an ideal world, autonomy in the operation of technical systems promises several advantages such as extended capabilities as well as reduction of the design and supervision efforts. Despite that goal, currently in practice, more often than not, it leads to extra burden on the design process and a shift of efforts to additional supervision of autonomic aspects. Moreover, current autonomous systems face other challenges too, trustworthiness and ability to provision hard guarantees among them. Lack of predictability of system operations, especially in corner cases, is also part of these challenges. This is in contrast to traditional system design and can range—particularly for functional tasks and capabilities—anywhere from undesired to prohibited. For example, in safety critical applications, providing guaranteed operations and performance is paramount and rather nonnegotiable. In many industrial applications, for example, a production line, some variations from an expected behavior may be acceptable (when they cause no considerable change in the quality of the final product), some may lead to product losses, and some may lead to safety issues. Therefore, variations in the behavior of such an autonomous system might be acceptable, undesired, or unacceptable. For extra-functional tasks, as we will discuss in the "Toward (more) dependable autonomy" section, these constraints are sometimes less pressing as they do not have to follow precise standards or specifications and, more often than not, any gains due to autonomy are highly appreciated. The focus of this section, however, is on the concerns regarding functional tasks and properties and their respective guarantees.

A traditional solution for providing guarantees for a system is using verification methods to test the behavior of the system under various conditions and determine the boundaries of behaviors and provide guarantees accordingly. However, the space of possible behaviors and conditions grows exponentially with the increase in complexity and autonomy. For example, testing an autonomous driving car for its behavior in every possible condition (caused by other cars, pedestrians, animals crossing the road, objects fallen on the road, as well as different weather and road conditions) is practically impossible. Consequently, providing full guarantees for such systems is also impossible. However, they can be partially tested and verified for a subset of possible conditions.

One could argue that providing such guarantees for a human driver is also impossible and therefore, expecting them from autonomous driving vehicles is not reasonable. However, we need to consider

Digital Object Identifier 10.1109/MDAT.2020.3024145
Date of publication: 14 September 2020; date of current version: 9 February 2022.

2168-2364/20©2020 IEEE

two factors. First, any mistake by an autonomous system is probably an indicator of a systematic error. That means that all such autonomous systems—which may add up to a very large number—may commit the same mistake with a high likelihood, whereas this is not the case for human drivers. That is one of the reasons why targeting trust and guarantees are so pressing.

Second, regulators and the public at large may not be quite ready yet for accepting that machines are allowed to fail stochastically. We cannot ignore the cultural element of such expectations, set—at least in part—by the predictable operations of the machines and systems designed and built from the beginning of the industrial revolution. In the Cyber-Physical European Roadmap and Strategy,[1] the European Commission explicitly mentions the societal challenges as one of the six main challenges to be overcome and emphasizes the importance of raising awareness and ensuring trustworthiness. Raising awareness and establishing trust normally take a rather longer time to prove effective and only in the long run may relax the requirements on autonomous system design.

In a different approach, the Information and Communication Technologies program of Horizon 2020 of the European Commission[1] invites researchers and innovators to "focus on autonomic solutions capable of guaranteeing the overall reliability and security even when the components or subsystems are not fully reliable and unforeseen conditions emerge in the course of operation."[1] Presenting dependability and reliability of the cyber–physical system (CPS) as one the main pillars of their development. Even though achieving this goal is complex and hard, it is reasonable to think that using current engineering approaches, such a vision can—to an extent—come to realization, probably earlier than cultural preparations for acceptance of unpredictable machine behaviors by public at large.

For these reasons, there is hesitation, even resistance, against fully embracing autonomous systems in many sections of industry too. Even though initiatives, such as Industry 4.0, have helped in creating a more open approach toward such techniques, often autonomy of the systems is significantly reduced to predetermined scenarios to provide guaranteed behaviors.

Toward (more) dependable autonomy

In systems engineering, dependability is considered as a measure for the availability, reliability, and maintainability of a system. These qualities can either be obtained in a deterministic manner, by design and prior to system deployment, and/or in a probabilistic manner during dynamic, autonomous control of a system at runtime. An (incomplete) selection of best practice approaches for combining autonomous control with improved dependability in system operation is subject of the following subsections. We consider improvements in dependable autonomy as a prerequisite for increasing trust.

Guarded guarantees

Here, we present one of the approaches which can provide certain guarantees and is compatible with autonomous behavior is using operational corridors. Working within hard-bounded corridors for operational parameters guarantees functional and extra-functional lower and upper bounds for system operations. For instance, in the context of a multiprocessor system-on-chip (MPSoC), supply voltage (V_{dd}), frequency (f), and number of time division multiplexed (TDM)-slotted tasks within work queue per core could be considered as these (hard-bounded) operational corridors, which guarantee performance (functional) and other characteristics of the system such as power dissipation and temperature (extra-functional). Furthermore, varying these parameters with relative step functions (e.g., increase/decrease by ±10%, or more conservative, ±5%) typically allows the identification of trends or at least the direction in which the system operation point is moving. Defining safety margins around the hard corridor bounds represent buffer zones in which corresponding counter actions toward current trends can be applied when individual components or subsystems of the autonomous system leave the target area of desired performance-power tradeoff, symbolically shown as the green area in Figure 1 [1]. In general, the safety margin (shown in yellow on Figure 1) shall be dimensioned such that multiple steps of varying an operation parameter would be necessary to cross the margin into a violation. This ensures that there are multiple opportunities for the autonomous system to apply counter actions and prevent violations. Depending on the application domain of an autonomous system, satisfying the minimum operation bounds must at least guarantee that the system gets into a safe halt (i.e., "fail-safe" through temporally limited guarantee of minimum performance bound) or continues operation at

[1] http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/ict-01-2019.html [Online since 2017; last access July 27, 2018].

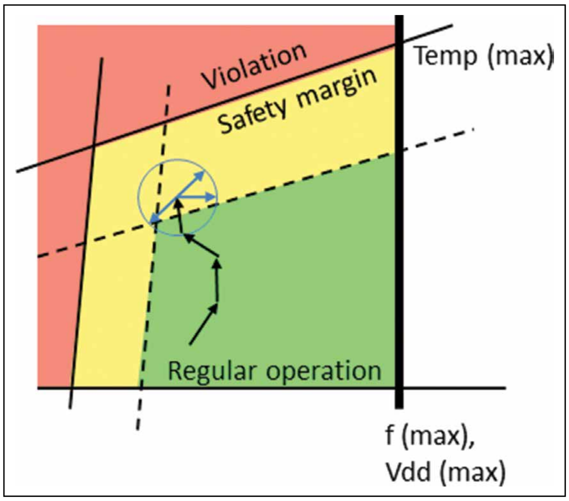

Figure 1. Tendency of the component or system operation to move into the safety margin zone with degraded behavior allows for corrective actions to prevent constraint violation possibly leading to a failure [1].

degraded but still acceptable performance (i.e., "fail-operational" or "limp home" strategy).

One way of implementing such strategies is using learning classifier-based reinforcement machine learning [2] which allows conducting autonomous control on the basis of predefined, pretested conditional rules which are not only human interpretable, but also can be specified such that they reflect best-practice human engineering knowledge and practice. This is an important property to increase trust in the applied control actions. Counterintuitive condition–action pairs or fitness values can be explicitly double-checked or eliminated without disrupting the remaining rule sets. This is an important conceptual difference to weight adjustments in (deep) convolutional neural network (NN).

Divide and conquer

Another approach that we recommend is classification of tasks into different criticality bins (e.g., best effort, real-time, and safety-critical bins). This enables the relaxation/constraining of application-specific decision-making. That is, the system can make decision in a self-aware fashion and depending on the criticality of respective tasks. For example, best effort tasks/applications may benefit from (and also tolerate) a more proactive "risky" behavior, whereas to critical tasks and applications conservative policies with firm guarantees shall be applied [3]. Prerequisite for applying different policies per criticality class is the availability of techniques for proper task isolation. Virtualization methods or hypervisors, which consequently separate processing, interconnect, I/O, and memory resources of different virtual machines, also known as task classes, are examples of such isolation techniques.

Orthogonality in design

Orthogonal redundancy is a form of redundancy where the backup device or method is completely different from the primary device or method that is prone to error. Therefore, the failure modes of the two devices or methods do not intersect with each other. This safeguards the total system against catastrophic failures. Here, analogous to orthogonal redundancy, we propose "orthogonal design." Orthogonality in design can, for example, be achieved through diverse spatial redundancy. Like conventional functional units, autonomous entities too can be deployed in redundant manner [e.g., dual modular redundancy (DMR) and triple modular redundancy (TMR)] and in form of different architectural realizations to avoid identical misbehavior. This characteristic, avoiding identical misbehavior, is also a mandatory prerequisite for provisioning fail-operational properties, as described in [4] for example. Of course, this comes at additional resource expenditure and hence should be reserved for critical tasks that truly demand it.

However, we believe this method can be used for less critical applications too. All system behaviors can, in general, be partitioned into intended functional duties (i.e., the functional specs) and extra-functional characteristics, such as power dissipation, temperature variation on a silicon chip, or printed circuit board, as well as environmental and/or manufacturing induced variability exposures (e.g., in deep submicron CMOS technologies). If extra-functional characteristics, which can be considerably important for the proper system operation, can independently be tuned and modified from the functional duties, one can consider functional and extra-functional requirements orthogonal to each other (see Figure 2). We define orthogonal autonomy as when either functional or extra-functional properties (or both) are autonomously controlled and optimized. In the context of autonomous systems, we particularly see a high potential in the autonomous optimization of extra-functional properties. Such optimization techniques may also include the exploitation of emergent behavior, which is known to be highly effective (when

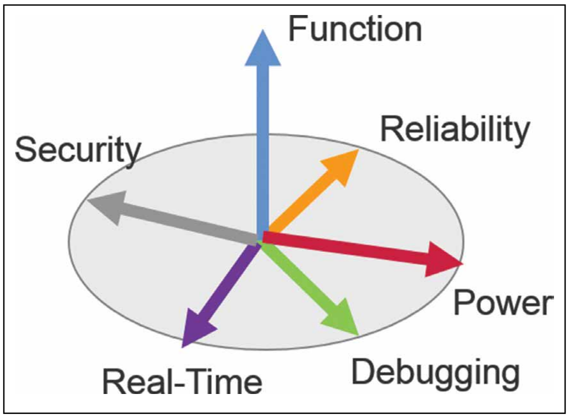

Figure 2. Orthogonal autonomy; complexity increases either due to functional requirements or consideration of pressing extra-functional aspects such as security, resilience, and power dissipation. Orthogonal autonomoy can improve the extra-functional aspects without jeopardizing the required functional guarantees.

applicable). Its applicability to the pure functional domain depends on whether the functional behavior, which very often follows industrial standards or human generated specifications, can be mapped onto a suitable emergent behavior, which is rarely the case. However, often explicit standards or guidelines for extra-functional properties, such as how to achieve better power efficiency or balanced temperature profiles, do not exist. Hence, any improvement obtained from autonomous emergent behavior on those aspects is highly appreciated.

An example of orthogonal autonomy is the early warning score (EWS) system presented in [5], where "attention" [6] is used to improve power efficiency of the system. To this end, the system adjusts the frequency of sensory data transmission and thus reduces the power consumption to half. However, this does not affect the functionality of the system which is the correct assessment of the health condition of the subject.

Expressive autonomous system

Establishing trust

As discussed in the introductory section, a major issue in adopting autonomous systems is the trust between the (human) user and the system (or machine). One way of increasing trustworthiness—which we advocate in this article—is by improving transparency and providing explanations and reasons behind the behaviors, decisions, and operations of an autonomous system. This does not necessarily improve the dependability of the system, however, it improves the predictability from the perspective of the user and paves the way for its wider adoption.

The need for machines to provide such explanations is widely acknowledged and its reflection can be seen in examples such as the explainable artificial intelligence (XAI) program of the Defense Advanced Research Projects Agency (DARPA).[2] The aim of XAI is to "understand, appropriately trust, and effectively manage an emerging generation of artificially intelligent machine partners."[2] To this end, the ideal is depicted as a machine that can respond (in a fashion understandable for its human users) to the following questions[2]:

- Why did you do that and not something else?
- How do I correct an error?
- When can I trust you?
- When do you succeed or fail?

To that we propose to add questions such as:

- How do I change a certain behavior of yours (regardless of its correctness)?
- What were other possible courses of action?
- What are probable consequences of each potential action?
- How do I add a new option/course of action?

Symbolic artificial intelligence (AI), in particular, between 1950s and 1980s aimed at similar objectives and explored some parts of the field [7]. However, many of currently main stream AI algorithms (such as NN) compromise explanation in favor of prediction [8]. Therefore, systems using them cannot provide explanations or justifications for their decisions. These systems can be represented by curves similar to A (blue) on the radar chart of Figure 3.

In some other systems, such as the ones represented by the B curve (orange) on Figure 3, even the accuracy of the system might be compromised to achieve the necessary speed requirements. One of the major issues with expressive system development is that the complexity of the system explodes exponentially when addition of explanation is considered, in particular,

[2] https://www.darpa.mil/program/explainable-artificial-intelligence [Online since 2016; last access July 23, 2018].

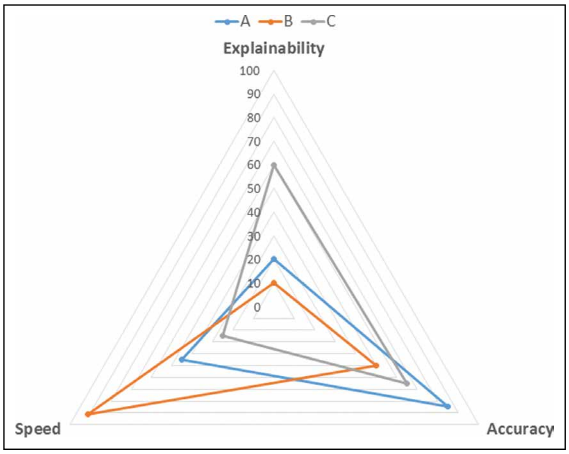

Figure 3. Compromise between accuracy, explainability, and speed of intelligent and autonomous systems. With a given set of resources, achieving a high score on each of the axes comes at the price of lower scores on one or both of the other two.

when this explanation is supposed to be presented in human language (or easily understandable by humans). That in turn leads to slow systems and it often compromises the accuracy of the systems too. Therefore, current explainable systems can be—at best—presented by curves similar to C (gray) on Figure 3.

The path forward, we contend, is to consider a combination of these techniques: first, the usage of alternative more expressive approaches such as probabilistic modeling [8]; second, coming up with new infrastructure and methods for adding explainability features (in particular, those understandable by human users) which do not require substantial compromise on accuracy of the systems and more importantly do not require massive computational resources. Although some machines can answer to some of those questions in limited capacities and more will be attained in the near future, such an ideal position in its full glory seems attainable only in the farther future.

Machine user cooperation

From the discussion in the introductory section, we can see that one of the main reasons for not adopting autonomous systems as widely as their potential suggests, is lack of trust. Particularly, lack of trust regarding the consequences of the actions these systems take (some of which may be undesirable). We argued that providing explanations can help users in understanding and accepting these actions better. However, we propose another remedy to this issue too, namely self-awareness of the autonomous system and machine user cooperation. For instance, when the system has a low “confidence” [9] regarding its decision, it could involve the user in the decision-making process. It has been shown [9], [10] that self-awareness of the system regarding its confidence can improve its performance in an autonomous fashion. However, that is not always the case. Therefore, in cases where the system does not reach to a conclusive and confident decision, in our proposed scheme, it could ask for additional user input or delegate the decision entirely to the user. In the latter case the system does not take any action by itself and cannot produce any negative effect. Since this also decreases their positive effect, this approach provides a suboptimal but short term solution for the near future.

We envision “taking no actions” at two different levels. In the first approach, the role of the system can be limited to supervision only and to notify the operator about decision points. It is then the operator who needs to evaluate the situation, make a decision, and take appropriate actions. In another approach, the system can perform extensive analysis and explore several potential actions and their probable consequences (when possible verified through simulations) and provide these information to the user. This minimizes the burden of the user while allowing the user to guide the system toward dependable behaviors in conditions where the autonomous system might have not behaved very desirably. For example, in continuation of the project Self-Aware health Monitoring and Bio-inspired coordination for distributed Automation systems (SAMBA) [11], it is planned to use the context-aware health monitoring (CAH) system [12] to monitor the system, analyze the situation of the production line, make decisions, and verify them through simulations (since no real action on the floor is permitted) and provide a list of suggestions to the operator. The operators can decide which option meets the goals the best, or pick another solution at their own discretion. Thus, this solution can lead to more dependable behaviors and increased trust for the user (due to improved behavior as well as user’s involvement in the decision-making process).

Discussions and conclusions

Truly autonomous systems require a high degree of flexibility but also need to provide guarantees for safety-critical functions and establish trust with human users. While many techniques exist, as partially reviewed above, a comprehensive solution for autonomous system design requires more research. The field of self-aware computing, that has emerged during the last two decades, can provide a key faculty by endowing an autonomous system with the capability of continuous, extensive self-observation, and self-assessment.

Self-assessment in autonomous systems

Although self-assessment is not necessarily a prerequisite for achieving autonomy, we contend that it is a fundamental necessity for expressive systems to be able to explain their behaviors and status. Similarly, we maintain that it is a prerequisite for machine user cooperation as sketched in the "Expressive autonomous system" section. By self-assessment or computational self-awareness here we refer to the ability of the system to observe its resources, behaviors, expectations, and goals. As such the system knows itself, how well it is performing, if it still provides guarantees, how likely it is to succeed and when it is lost. In a higher level of awareness, the system is able to assess its environment (what is external to it) and situate itself with respect to its environment. Such information can enhance its decision making and enables it to explain itself to the user or provide it with options and necessary information regarding those options. Moreover, even though it is not a prerequisite to autonomy, it can enrich the autonomy of the system by helping it in navigating the decision space since

- it can reflect on its own actions and assess what guarantees would be still valid in any given case;
- it can know how confident it is in pursuing a line of action and how it may affect required guarantees;
- it can predict some of the cases which may lead to violation or leaving its scope of operation.

We note that self-awareness is by nature orthogonal to the functions of the system, however, it can affect the functional path, if its outcomes are used for decision making of the system regarding its functional operations.

Even though self-awareness and self-assessment in computing systems have been studied extensively, we believe there is a large opportunity for working on what should be done in the context of safety-critical systems to support the kind of autonomy discussed in this article.

THE RAPID DEVELOPMENT of machine learning and other AI technologies gradually facilitate more autonomous decision making of systems in complex, dynamic environments. While this is laudable because it promises to increase safety and comfort at reduced design, operation, and maintenance costs, we do observe a tension between increased autonomy on one hand and a reduced level of hard guarantees and trust on the other hand. The complex AI techniques that hold these promises are often probabilistic in nature, cannot provably demonstrate that all corner cases are covered, and their effects are exceedingly hard to understand and predict by human users.

Table 1. Summary of approaches discussed in this work and their effect on the autonomy of the system, the guarantees it can provide, and the trust of the user in autonomous systems.

		Effect on	
Approach	Guarantees	Trust	Autonomy
Raising awareness in society	-	↑	-
Guarded guarantees	↑	↑	↓
Criticality bins	↑	-	-
Orthogonal autonomy (extra-f.)	-	-	↑
Explanation of behaviors	-	↑	-
Machine user cooperation	↑	↑	↓
Computational self-awareness	↑	↑	↑

We argue that remedies for both concerns, limited guarantees and limited trust, have to be found. The state of the art offers some solutions in both areas and we foresee some solutions with good potential, which we have presented in "Toward (more) dependable autonomy" and "Expressive autonomous system" sections. A summary of these approaches and how they affect each of the main three factors discussed here (i.e., autonomy, guarantees, and trust) is provided in Table 1. However, we concede that in practice much still needs to be done. In particular, we observe that autonomous systems have to be keenly aware of their capabilities and their limitations. They have to know, when they can provide safety guarantees, when their offered line of action will most likely lead to a safe and comfortable experience, and when they are at a loss facing unknown territories. Based on more accurate and reliable self-assessment, autonomous systems can explain their actions, and thus gain the trust of users who would then better understand what guarantees are provided and what the limitations of the system are. ■

Acknowledgments

We would like to thank organizers and attendees of the Self-aware cyber–Physical System (SelPhyS) workshop for providing an invigorating environment for discussions about self-aware and autonomous systems, which led to the initial idea of this article. We like to further acknowledge the fruitful collaboration with partners from the University of Tübingen (Prof. Rosenstiel), TU Braunschweig (Prof. Ernst), UC Irvine (Profs. Dutt and Kurdahi), and KIT Karlsruhe (Prof. Becker) on previous and current projects by the DFG (German Research Foundation) within the Priority Program 1183 Organic Computing, the Grant HE4584/7-1 Information Processing Factory, as well as on the BMBF Project ARAMiS. We also acknowledge the financial support from the Austrian Government through BMVIT/FFG in the Project SAMBA (FFG 855426) under the *ICT of the Future* program and the project SAVE (FFG 864883) under the *Production of the Future* program, and through BMDW/Christian Doppler Research Association in the CD-Lab on Embedded Machine Learning.

References

[1] J. Zeppenfeld et al., “Towards scalability and reliability of autonomic systems on chip,” in *Proc. 13th IEEE Int. Symp. Object/Component/Service-Oriented Real-Time Distrib. Comput. Workshops*, Seville, Spain, May 2010, pp. 73–80.

[2] S. W. Wilson, “Classifier fitness based on accuracy,” *Evol. Comput.*, vol. 3, no. 2, pp. 149–175, Jun. 1995.

[3] A. Sadighi et al., “Design methodologies for enabling self-awareness in autonomous systems,” in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Dresden, Germany, Mar. 2018, pp. 1532–1537.

[4] A. Kohn et al., “Fail-operational in safety-related automotive multi-core systems,” in *Proc. 10th IEEE Int. Symp. Ind. Embedded Syst. (SIES)*, Siegen, Germany, Jun. 2015, pp. 144–147.

[5] A. Anzanpour et al., “Self-awareness in remote health monitoring systems using wearable electronics,” in *Proc. Design Autom. Test Eur. Conf. Exhibit. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 1056–1061.

[6] N. TaheriNejad, A. Jantsch, and D. Pollreisz, “Comprehensive observation and its role in self-awareness; an emotion recognition system example,” in *Proc. Position Papers Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Oct. 2016, pp. 117–124.

[7] John Haugeland, *Artificial Intelligence: The Very Idea*. Cambridge, MA, USA: MIT Press, 1985.

[8] M. D. Blei, “Expressive probabilistic models and scalable method of moments: Technical perspective,” *Commun. ACM*, vol. 61, p. 84, Mar. 2018.

[9] N. TaheriNejad and A. Jantsch, “Improved machine learning using confidence,” in *Proc. IEEE Can. Conf. Electron. Comput. Eng. (CCECE)*, May 2019, pp. 1–5.

[10] H. A. Kholerdi, N. TaheriNejad, and A. Jantsch, “Enhancement of classification of small data sets using self-awareness—An iris flower case-study,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[11] L. C. Siafara et al., “SAMBA—An architecture for adaptive cognitive control of distributed cyber-physical production systems based on its self-awareness,” *e&i Elektrotechnik und Informationstechnik*, vol. 135, no. 3, pp. 270–277, Jun. 2018.

[12] M. Gotzinger et al., “On the design of context-aware health monitoring without a priori knowledge; an AC-motor case-study,” in *Proc. IEEE 30th Can. Conf. Electron. Comput. Eng. (CCECE)*, Apr. 2017, pp. 1–5.

Nima TaheriNejad is a Universität-assistant with TU Wien (formerly known as Vienna University of Technology), Vienna, Austria. His research interests include self-awareness in resource-constrained cyber–physical (embedded) systems, memristor-based circuit and systems, and health-care. TaheriNejad has a PhD in electrical and computer engineering from The University of British Columbia (UBC). He is a member of ACM and the IEEE Circuits and Systems Society as well as the IEEE Engineering in Medicine and Biology Society.

Andreas Herkersdorf is a Professor with the Department of Electrical and Computer Engineering and the Department of Informatics, Technical University of Munich (TUM), Munich, Germany. His research interests include application-specific multiprocessor architectures, IP network processing, and self-adaptive fault-tolerant computing. Herkersdorf has a PhD from ETH Zürich. He is a member of the German Research Foundation (DFG) Review Board and a Senior Member of IEEE.

Axel Jantsch has been a Full Professor of Systems on Chip with TU Wien, Vienna, Austria, since 2015. His research interests include on embedded machine learning and self-aware cyber–physical systems. Jantsch has Dipl. Ing. and Dr. Tech. from TU Wien.

■ Direct questions and comments about this article to Nima Taherinejad, Institute of Computer Technology (ICT), TU Wien, 1040 Vienna, Austria; nima.taherinejad@tuwien.ac.at.