IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 8, 2019, accepted November 21, 2019, date of publication November 26, 2019,
date of current version December 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956124

Self-Adaptation Techniques in Cyber-Physical Systems (CPSs)

SHERALI ZEADALLY[1], TEODORA SANISLAV[2], (MEMBER, IEEE), AND GEORGE DAN MOIS[2], (MEMBER, IEEE)

[1] College of Communication and Information, University of Kentucky, Lexington, KY 40506, USA
[2] Automation Department, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania

Corresponding author: Teodora Sanislav (teodora.sanislav@aut.utcluj.ro)

ABSTRACT Cyber-Physical Systems (CPSs) are large-scale complex systems that monitor and control physical processes by using computer algorithms tightly integrated with networking and their users. Monitoring and controlling the physical environment is a hot topic for today's researchers and engineers in academia and industry. Within this realm, an important feature of current and future Information and Communications Technology (ICT) systems is self-adaptation—yet there is a shortage of information focusing on this characteristic in the literature, particularly as it relates to CPSs. Here, we investigate current state-of-the-art research on CPSs from this perspective, and evaluate the main self-adaptive approaches proposed in the literature, along with their results, strengths, and weaknesses. We also discuss appropriate techniques for enabling self-adaptation capabilities within CPSs at different architectural layers. Overcoming the challenges associated with designing and implementing self-adaptive mechanisms in CPSs will provide a path for bolstering a new generation of CPSs with greater robustness and reliability.

INDEX TERMS Adaptive systems, cyber-physical systems, self-adaptation.

I. INTRODUCTION

Cyber-Physical Systems (CPSs) refer to a new generation of engineered systems where cyber and physical components are strongly interconnected, each operating on different spatial and temporal scales, exhibiting multiple, distinct behaviors, and interacting in numerous ways that change depending on context [1]. CPSs aim to exceed the capabilities of traditional large-scale systems (such as power grids [2] or today's automation solutions in production systems [3]) in terms of adaptability, autonomy, efficiency, functionality, reliability, safety, scalability, and usability, making them more precise and highly efficient. CPSs can operate in dangerous or inaccessible environments (including search and rescue, firefighting, planetary surface exploration, and deep-sea exploration) to provide large-scale and distributed control, which enhance human capabilities and quality of life [1], [4], [5]. The US National Science Foundation (NSF) first introduced term *Cyber-Physical Systems* in 2006. Since then, many variations of its definition have emerged [1], [6]–[8], all of which underscore that CPSs possess the following characteristics [9]: cyber capabilities in every physical component, networked on an enormous scale; dynamic reconfiguration; a high degree of automation; dependable operations; self-organization; and both cyber and physical components integrated for learning, self-adaptation, and higher performance.

These aforementioned CPS characteristics recently became the subject of intensive research efforts, fueled by the challenges and requirements that arise when implementing large-scale, complex systems in the following domains: transportation and mobility (autonomous and smart vehicles, interactive traffic control systems), energy (electricity systems, renewable energy supply systems, and smart oil and gas distribution grids), civil infrastructure (water and wastewater treatment systems, monitoring and control systems, and early warning systems), environment (monitoring and control systems, emergency response systems), healthcare (smart medical devices, assistive systems, and disease diagnosis and prevention systems), buildings (building automation systems), defense (smart weapons, intelligent unmanned vehicles), manufacturing and production (supply chain and logistics systems, robots), agriculture (irrigation systems), and many others [10]. Thus, CPSs' development requires transdisciplinary approaches and emerging technologies such as Artificial Intelligence (AI), big data, cloud computing,

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Yu.

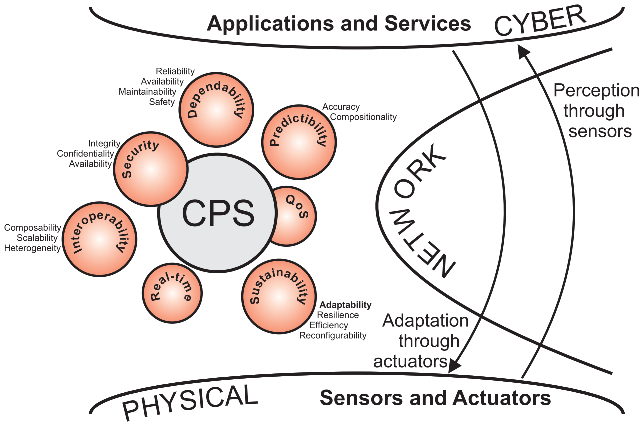

FIGURE 1. General overview of Cyber-Physical Systems (CPSs) and their challenges.

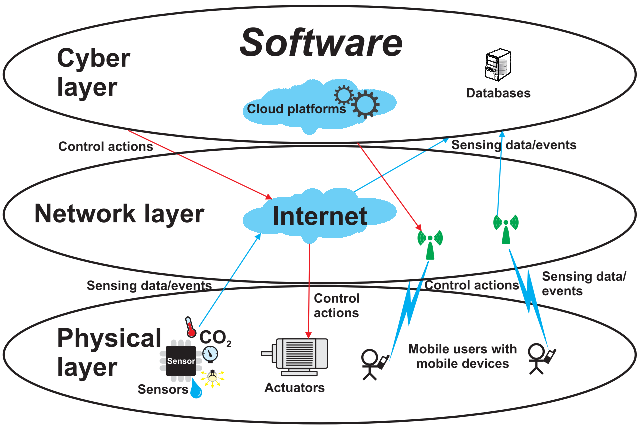

FIGURE 2. Architectural layers of CPSs.

the Internet of Things (IoT), next-generation robotics, and distributed manufacturing. CPSs will bring important benefits to society, including safer, more efficient, and more reliable infrastructures in various application domains, increased energy efficiency, improved quality of life, and enhanced global competitiveness.

Currently, though, advances in CPSs are hindered by significant research challenges that both academia and industry must address. These challenges include CPS-related aspects of design and modeling (architectures, models, tools, and programming frameworks) [11]; security [12], dependability, interoperability, predictability, and sustainability [13]; Quality of Service (QoS) issues [14]; and real-time requirements [15]. In particular, the security aspect deals with integrity, confidentiality, and availability attributes, while the dependability aspect covers reliability, availability, maintainability, and safety attributes. Another important challenge is interoperability, which must manage CPSs' composability, scalability, and heterogeneity attributes. Predictability involves accuracy and compositionality attributes, while sustainability transpires through adaptability, resilience, reconfigurability, and efficiency characteristics [16]. Figure 1 depicts these challenges within a general overview, and highlights the three main architectural layers: physical, network, and cyber. Figure 2 further describes the layers and their interactions; the physical layer is composed of sensors, actuators, and systems based on embedded processors; networking facilitates the communication between components in the CPS; and cyber includes applications and services.

Recent literature contains various surveys that focus on presenting a holistic view of CPSs. In Shi *et al.* [17], they briefly introduce CPSs' features, research challenges, and applications. Gunes *et al.* [16] present CPSs' history, applications, and related issues, along with concepts similar to CPSs—e.g., System of Systems (SoS), IoT, big data, Wireless Sensor Networks (WSNs), and current research efforts in various application domains to achieve the CPSs' vision. In Khaitan and McCalley [18], they survey several recent research efforts in the field of CPSs. They classify these efforts into different CPS-related categories, based on whether they deal with design and development, address specific issues, or discuss applying CPSs in specific domains, and identify future research challenges. They also present a wide range of CPS examples (such as modern vehicles [19], [20], medical and healthcare systems [21], and smart homes and buildings [22]). The challenge of analyzing and defining the CPSs' characteristics specific to different application domains also is addressed in the literature. In Yu and Xue [2], they present an overview of the technological challenges faced by smart grids in the context of CPSs (architecture and abstraction; communication technologies; modeling and simulation; cybersecurity; and distributed computation, optimization, intelligence, and control), as well as the implications of current technological advances in these state-of-the-art smart grid networks. In Jia *et al.* [23], they present a comprehensive survey on platoon-based Vehicular CPSs (VCPSs), highlighting fundamental issues such as cluster management, cooperative driving, and system communication.

In short, many recently published papers on CPSs presented overviews of CPSs' designs and architectures, and focused on related fundamental aspects such as functionality, performance, security, reliability, and scalability [2], [16], [18], [23]–[25]. They also discussed future research opportunities in the areas of cybersecurity and the IoT. In contrast to most previously published papers on CPSs, one important area that has not received enough attention is self-adaptation in CPSs. The systematic literature reviews in Muccini *et al.* [26] is one of the few studies that investigated the role of self-adaptation within CPSs. The authors statistically analyzed several approaches proposed to implement self-adaptation solutions, which combine different adaptation mechanisms within and across the layers of the technology stack (e.g., the application, middleware, communication, service, and cloud layers). They found 42 research papers that focused on self-adaptation out of a total of 1,103 CPSs papers from four major scientific databases (IEEE Xplore, ACM Digital Library, SpringerLink, and ScienceDirect).

A. CONTRIBUTIONS OF THIS WORK

In contrast to the work presented in [26], here we not only carefully analyze each solution that addresses the self-adaptation issue, but we discuss the main techniques that can be included into a general architecture of CPSs to ensure self-adaptive capabilities for each of the three main layers (physical, network, and cyber) of this category of systems.

We summarize the contributions of this work as follows:

- We review state-of-the-art results to date in the area of self-adaptation in CPSs. In particular, we focus on recently proposed self-adaptation mechanisms and identify their strengths and weaknesses.
- Based on a comprehensive review of self-adaptive solutions for CPSs, we discuss different techniques for implementing self-adaptive schemes at each CPS's architectural layer (physical, network, and cyber).

B. ORGANIZATION

The rest of this paper is organized as follows. Section II describes self-adaptation techniques in CPS, and discusses related issues associated with this system characteristic. Section III presents self-adaptation approaches proposed at CPSs' architectural layers, along with the challenges that must be addressed to reap self-adaptation benefits. Section IV discusses self-adaptation mechanisms that can be applied at each layer of CPSs for ensuring the expected behavior of these systems. Finally, we provide concluding remarks in Section V.

II. CPS SELF-ADAPTATION AND RELATED ISSUES

Adaptation is an essential property of living systems. This concept, defined in general terms, is a process that is about modifying something (itself, others, or the environment) so as to make it better-suited for some purpose than it would be otherwise [27]. This term has been used frequently in biology since the 19th century, and later it was borrowed by other disciplines such as psychology, economics, and engineering. In engineering, the definition of *adaptability* refers to a system's ability to cope with threats while maintaining its parameters within certain previously set reference values [27]. Examples include maintaining communication latency and reliability within a smart grid when subjected to increased additional communication and processing loads caused by Denial of Service (DoS) attacks [28]; and maintaining request-response latency below some threshold within a client-server system through software strategies when responding to high-latency situations [29].

Significant advances in the engineering of both hardware and software systems have led to the emergence of the notion of self-adaptation. The "self" prefix indicates that the adaptation action is triggered autonomously by the system in question to adjust to changes that occur in its context and environment. Scientific literature provides several definitions of self-adaptive systems. For example, some define a self-adaptive system as a system that can change its behavior and structure to adapt in response to changes in itself or in its operating environment [30]. In 1997, the US Defense Advanced Research Project Agency (DARPA) referred to self-adaptive software as one that evaluates its own behavior when the evaluation shows that it cannot achieve what the software was supposed to do, or when improved functionality or performance is possible [31]. Adaptive mechanisms and self-awareness have also been applied to hardware [32], [33]. Razor, a general-purpose processor, reduces its operating voltage until it produces an erroneous result [32]. When an error is detected, the design—which includes the logic for recovery and correction—takes action to adjust the operating voltage and compute the correct result. Ultimately, this built-in self-adaptive mechanism that adapts the power supply to the error rate leads to significant energy savings.

The necessity of self-adaptation arises from the fundamental requirements that CPSs must meet. CPSs must be context-aware, partially or fully self-managed, and therefore make use of relevant services to capture the environment status and operating conditions to assure the expected behavior in all scenarios [34]. They also must evaluate their state and conditions autonomously and adapt reactively as quickly as possible.

CPSs are inherently feedback systems, because besides assuring predictability, and depending on the application domain, they must be adaptive. This self-adaptive characteristic can be in any of the three architectural layers (physical, network, and cyber), and can be implemented through actions taken based on simple logical reasoning or driven by knowledge extracted from big data analytics. For example, a wireless sensor can modify the rate of reporting (depending on its power supply) simply by increasing or decreasing the sleep time interval when the energy level exceeds some predefined threshold. On the other hand, a CPS's decision system concerned with environmental monitoring dynamically can change the rates of reporting for a large number of sensors, thereby adapting the system's operation, if—after analyzing the data—it finds that some data are unreliable or irrelevant. In this context, designing novel and complex systems must consider issues related to self-adaptation in CPSs, which include efficiency and performance, flexibility, reliability, configurability and reconfigurability, functionality, interoperability, dependability, security, scalability, composability, mobility, and self-adaptation techniques (software and hardware).

The self-adaptation techniques deployed in the design of CPSs often use the Monitor-Analyze-Plan-Execute (MAPE) model [35], which includes the feedback control loop, agents, self-organization techniques, control theory methods, and possible combinations [35], [36]. The MAPE model [35], [36] contains four important components with well-defined roles [36]. The first one is the Monitor component, which provides mechanisms for collecting, aggregating, filtering, and storing information gathered from the environment through sensors into a database. The second, the Analyze component, provides mechanisms that check whether an adaptation is required, and if so, triggers the Plan component that initiates an action plan (e.g., selecting the adaptation policies) needed to achieve the system's goals. Finally, the Execute component controls execution of the selected adaptation plan through actuators. Instead of using a standard database, this model uses a Knowledge database to store collected data, along with adaptation goals or other states shared by all the MAPE components. Agents and Multiagent Systems (MASs) represent another paradigm that could be quite useful in implementing feedback control loops to achieve self-adaptability in engineering large-scale, software-intensive systems [37], [38]–[40]. MASs' benefits stem from their fundamental characteristics (e.g., communication, negotiation, and learning), because they are goal-oriented software approaches capable of autonomously adapting to their environment at runtime. Additionally, the primary MAS characteristics such as autonomy, cooperation, reactivity, and proactivity make MASs suitable as entities capable of reacting to external events by adapting their behaviors and making informed decisions to fulfill tasks. Self-organization can be defined as a system's ability to arrange itself autonomously and spontaneously, mainly because of internal interactions, and without the need to use a central authority [41]. This form of adaptation can be applied, in combination with MASs, at different levels of granularity: at the micro level (the behavior of the CPS's low-level entities is adjusted), or at the macro level (the entire CPS architecture is modified). In Barbosa *et al.* [41], they noted that, at the micro level, self-organization can respond smoothly to perturbation, while at the macro level, different types of triggers (events that disrupt or deviate the system's predicted operation, such as a resource malfunction or a production quality issue in a manufacturing system) can cause drastic responsive actions—for example, an entire structural self-rearrangement of the CPS. Control theory can be used to establish and refine the adaptive actions, depending on feedback received from the sensors present in CPSs. The control applications residing in the cyber component of a CPS control the dynamic systems' behavior, which belong to the physical part by generating command actions based on sensor data and mathematical models, and by transmitting them through the network to the actuators. The generic mechanism for achieving self-adaptation in this case relies on the following: collecting data about the system state; enacting changes in the environment and context (changes in the system's state); analyzing and extracting relevant information, such as trends or symptoms; decision making; and finally, taking adaptive action [42]. In the context of control theory, adaptive control represents a particular variation of control where the mathematical model used or the control law are adjusted to better respond to changes in the physical processes being managed [42]. The adaptive control strategy can support CPSs, because they are highly distributed and heterogeneous systems operating in continuously changing environments. For instance, adaptive control mechanisms can be used to stabilize these systems in the presence of uncertainty and attacks. Model Identification Adaptive Control (MIAC) [43], [44] and Model Reference Adaptive Control (MRAC) [43], [44] can cope with changes of the controlled process by installing additional feedback control loops that are excellent mechanisms to handle uncertainties [45]. In the context of a CPS, we can use various combinations consisting of the aforementioned techniques (i.e., MAPE and agents; self-organization and agents; MAPE; and MAS and self-organization) to improve the self-adaptation feature, because CPSs are heterogeneous, have no centralized control, and span over large areas.

III. ANALYZING CPS SELF-ADAPTATION SOLUTIONS

Different self-adaptation approaches in CPSs have been proposed in recent years in various application areas. In this section, we discuss some of the proposed solutions in terms of their results achieved, strengths, and weaknesses.

Kit *et al.* [46] proposed a model and framework for developing complex smart CPSs, built on the concepts of autonomous components and ensembles, and demonstrating the model's use in a parking scenario, where vehicles or autonomous components are equipped with vehicle–vehicle communication capabilities and sensors that enable them to share information and detect free parking spaces. This model operates at the middleware layer in the CPS design architecture. Each model component stores information about available parking spaces, along with real-time processes (e.g., positioning) that calculate the car's current position and detect free parking spaces in its immediate vicinity. The processes in the presented model are time-triggered or event-triggered. By enabling the model's components to share knowledge between them, the proposed framework implements the CPS's control and adaptation logic through a MAPE over shared Knowledge (MAPE-K) loop. The components are grouped into ensembles (groups of components working together to achieve a common goal). Each component can act either as a coordinator (the first member of an ensemble) or as a member (component that assumes the member role with respect to a coordinator) [47]. The information exchange and knowledge sharing about available parking spaces is represented by assigning knowledge between an ensemble's components (from members to coordinators and vice versa). The proposed model's authors stated that it is suitable for use by different adaptation policies, because they used the "model@run.time" approach [48] in their implementation. The "model@run.time" approach manages complexity in runtime environments by developing adaptation mechanisms that leverage software models [48]. The proposed model was implemented by two runtime simulation frameworks written in C++ and Java. The Java simulator provides the opportunity for performing the CPS's experiments with a decentralized behavior, and it is integrated with a well-known network simulator—Objective Modular Network Testbed in C++ (OMNeT++)—to obtain estimations about network latency depending on parameters such as topology, location of communicating components, and communication reliability. The Java simulator also provides a reasoner that offers self-adaptation capabilities to the CPS during its design phase, so that it can react to environmental changes at runtime. The reasoner uses the CPS's goals and requirements to describe the appropriate system state at any time during its operation. The C++ runtime framework is concerned with the actual deployment on physical devices and is not detailed in the presented paper.

Zhu *et al.* [49] proposed a new approach for supporting the management and control of transportation systems with self-adaptive capabilities. This approach combines several emerging technologies, such as agent control, social signaling, and the IoT. This research effort highlights the design of an artificial cyber system as a replica of the physical transportation system. The artificial cyber system represents the CPS's physical layer, which integrates empirical and data traffic models. These models were developed after they were synthesized, considering all categories that include engineering, social, human, and environmental factors that could affect the cyber system. The interactions of the physical layer elements and their evolution in normal and abnormal situations are mapped into rules executed by the artificial system. The actual and artificial systems' outputs are analyzed through parallel execution, resulting in a hardware-in-the-loop system wherein interactions between actual devices and artificial software modules take place. By comparing and analyzing the two systems' behaviors, predictions about future traffic conditions, and adjustments of the management and control are obtained for both systems.

Manic *et al.* [50] presented an overview of issues associated with what they consider to be an important component of current and future CPSs, such as Building Energy Management Systems (BEMSs). Some of these BEMSs involve managing an extremely large number of heterogeneous sensors and actuators, as well as frequent internal and external changes concerning the smart buildings. A BEMS also should address its occupants' comfort, taking into account energy efficiency. These aforementioned issues require a new approach to design smart buildings in the context of CPSs. This approach should enable adaptability, multisensor fusion, modeling, dynamic optimization, and use of Computation Intelligence (CI). CI deals with Artificial Neural Networks (ANNs), Fuzzy Logic (FL), and evolutionary algorithms capable of extracting a generalized system behavior and adapting this behavior when uncertainties occur. The authors also present a case study—a thermal energy storage system for cooling—whose predictive control is supported by three components: the design of a Building Power Requirement (BPR) prediction module, a Utility Load Prediction (ULP) module, and an ANN-based controller. The BRP module provides predictions on the future power requirements of a building to the ANN controller, whereas the ULP module provides predictions of the utility's expected load for a subsequent time interval. Finally, the ANN controller controls the energy used by the thermal energy storage at each time step. Therefore, four factors (that include the total cost of cooling, the money lost due to exceeding a building's energy requirements, the amount of power, and the difference between the preset and actual temperatures) are kept to a minimum. The proposed approach demonstrates that CI techniques can be deployed to provide self-adaptive support within a CPS in the field of smart buildings, and leads to a significant improvement in the system's overall performance.

Another domain that requires attention is the smart grid domain. In this environment, we have various interconnected complex physical networks within the power network infrastructure, and cyber systems represented by sensors, ICT, and advanced technologies [2]. Nowadays, the modern power grid is transforming into a continuously evolving CPS with the emergence of various types of power-generation methods, Phasor Measurement Units (PMUs) for long-distance transmission networks, smart meters, smart houses, and others [51]. Consequently, this complex system faces a wide range of specific technological challenges (including vulnerability to cascading failures [52] and mitigating cyberattacks [53], increased efficiency, and reducing the carbon footprint). Within the smart grid, self-adaptation, self-organization, and self-learning mechanisms are required so that the entire system reacts properly to faults, attacks, and emergencies, resulting in resiliency, security, and safety. MAS technology has proven useful in implementing these characteristics, by enabling distributed intelligence and optimization [54], [55]. However, MAS technologies face a broad range of challenges themselves. These challenges are even harder in the smart grid environment, leading to tremendous difficulty in automating operations not only at micro levels (such as managing a micro grid), but also on a global scale (e.g., allocating resources in case of outages or blockages) by considering economic and social aspects. Developing new intelligent, adaptive, and robust control strategies for the global stability of these large-scale systems—exposed to countless uncertainties and disturbances—also must be considered, because existing approaches cannot contend with the increased complexity and magnitude of today's power grid. Current solutions for providing self-adaptation characteristics to smart grids through distributed control include the division of the system and the implementation of fine-grained feedback loops while considering their global impact on the electric grid and distributed model predictive control. The major drawback of these decentralized control methods is their inability to efficiently model large-scale CPSs [2].

The authors of [51] present a software system residing in the cloud that supports Dynamic Demand Response [56], particularly the detection and pre-emptive correction of the supply-demand mismatch by initiating demand-side management from consumers that ultimately will be scaled to the city of Los Angeles. The cloud-based platform consists of a semantic information integration pipeline, for gathering real-time data from sensors and different data sources, a secure repository for data sharing, scalable machine-learning models for predicting demand, and a Web portal and mobile application for visualization. This form of adaptation, implemented through a complex feedback loop residing in the cloud, demonstrates that intelligent and sustainable management can be achieved with the help of cloud computing and data-driven analytics in the CPS domain.

In [57], the authors investigated using contextual information for adapting complex systems' behavior, operating in open and non-deterministic environments. This contextual information includes the knowledge required for a mobile communication device for selecting the proper device to perform a computation or to send data or location and computing nodes' availability, as well as costs of the network links for a communication device. The authors argued that context awareness and autonomicity are indispensable in managing the complexity of state-of-the-art software-intensive CPSs. Communication and networking are major components within a CPS that can benefit from these characteristics, for ultimately adapting their operations in response to changes in the environment, the system's internal state, and the users' and applications' new needs. The issue of context-aware autonomic computing and communications is addressed in the context of the IoT, but it is similar to the one also required by a CPS. The heterogeneity and components with both the IoT and CPS are similar, starting from miniature devices such as Radio-frequency identification (RFID) tags, sensors, and actuators, and ending with more advanced systems that are possibly equipped with embedded parallel processors. The proposed model is validated by simulating a scenario in which autonomous and self-aware nodes are used to monitor and control the level of hazard in an area where a disaster has occurred. The system also includes Mobile Agents (MAs) that can be used for restore and rescue operations (e.g., rescuing injured people or performing safe restore operations after a disaster occurs, such as wide-scale fires). The authors showed that by employing self-adaptive and self-organization mechanisms, a reduction in the number of active MAs in the system can be supported without notable performance degradation.

In Wang *et al.* [58], they addressed the current status and advances of CPSs in manufacturing, showing their potential as components of future factories. As products become more complex, the production processes must adapt accordingly and respond to the needs of society for reaching sustainability (reuse of resources, energy efficiency, self-organization, and self-maintenance). The authors discussed using self-adaptation techniques such as agent technology and self-organization in implementing manufacturing systems. They highlighted the drawbacks of the multiagent control approach [59], because of its nondeterministic nature, and mentioned the use of the Holonic [60] and Evolvable Systems [61] paradigms or approaches based on Agent-oriented Architectures (AoA) as solutions that can overcome these barriers. Holonic manufacturing systems are a manufacturing paradigm where holons, which are autonomous self-organizing units that can communicate with other holons, assist the operator in controlling the system by selecting appropriate parameter settings. The holons within these systems find their own strategies and build their own structure for achieving their goals [62]. The Evolvable Production Systems paradigm assumes the coexistence of several independent process-oriented modules that have the capability to dynamically adapt to changing operating conditions [61]. In Van Brussel *et al.* [60], they also presented several examples of CPSs in the manufacturing sector. One of the examples, Festo's MiniProd, particularly depicts CPSs' adaptability characteristic [58]. The implementation of the CPS is based on an AoA approach, which ensures multiagent control through four types of agents—namely, machine resource, coalition leader, transportation system, and human-machine interface (HMI) agents. To guarantee the entire system's self-adaptation, the agents are supported at the physical layer by control boards especially designed to run a multiagent setup and communicate via different protocols and standards (Ethernet, RS232/RS485, and others). This form of CPS adaptation demonstrates that using intelligent controllers within manufacturing processes can cope with uncertainties specific to the manufacturing domain.

In Afanasov *et al.* [63], they proposed a context-oriented approach to develop self-adaptive software components for resource-constrained CPSs capable of dynamically adapting to unpredictable situations. The approach defines two conceptual notions, context and context group, to organize the possible states of a CPS and their combinations. These notions can be used in the design phase of a resource-constrained CPS. The context describes the system's behavioral variations associated with a particular situation, and it enables software components to adapt the system. The contexts are grouped based on common characteristics. The authors also use the concepts of context and context groups to develop ConesC—a context-oriented extension for nesC; nesC is an event-driven programming language compatible with the TinyOS platform designed to run on embedded devices in wireless sensors networks [63]. A preliminary evaluation of the proposed approach was conducted based on coupling and cohesion, evolving the software, and system overhead. The proposed approach was implemented for a wildlife-monitoring application. When the ConesC implementation was compared with a functionally equivalent nesC implementation, results showed that the former is more decoupled and cohesive than the traditional nesC implementation, and any modification of the application design can be performed with little effort, because the implementation is highly modular. As a result, in these conditions, the runtime system overhead is small. The proposed adaptation solution improves the quality of the resulting implementation in terms of testing, maintenance, and evolution.

There are other mechanisms that have been proposed to ensure self-adaptation in CPSs. These include meta-adaptation strategies [64], adaptive Petri nets (APNs) [65], and runtime-efficient probabilistic model checking based on Discrete-Time Markov Chains (DTMCs) and Probabilistic Computation Tree Logic (PCTL) [66]. In Gerostathopoulo *et al.* [64], they introduced the concept of meta-adaptation strategies, which applies the generation of a set of tactics produced at runtime to reflect changes in the environment. These strategies evaluate the generated set of tactics using a metric that ranks them based on their possible influence on the corresponding CPS. Two examples of meta-adaptation strategies are proposed, by which two different sets of tactics are obtained. These sets of tactics are generated using two strategies, namely Knowledge Exchange by Data Classification (e.g., approximating the values provided by a malfunctioning sensor in a monitoring system) and by Process Period Adjustment (e.g., time scheduling of a duty-cycling system when timing violations occur). In the use cases presented in the paper, the first set of tactics estimates the values of collected data within a CPS to compensate the missing values caused by a sensor's failure, while the second set of tactics improves the collected data's accuracy by optimizing process schedules within complex systems. Since the proposed meta-adaptation strategies involve studying a CPS at runtime, they have higher potential than predesign strategies, but they also require a dedicated hardware infrastructure for analyzing the collected time series. In the second case, reducing processing times may affect other resources such as battery, network, and CPU usage of the CPS. In Ding *et al.* [65], they proposed using an extension of hybrid Petri nets for modeling a self-adaptive software system. The extension of the hybrid Petri net is achieved by including a neural network algorithm at specific transitions results in an APN. The proposed system's adaptive property stems from the underlying neural network's learning capacity. The authors used a manufacturing process to demonstrate the proposed solution's superiority over traditional optimization approaches. This superiority comes from the capability of the proposed system to make decisions based on runtime data and model the software system's behavior. In Filieri *et al.* [66], they focused on changes in the environment (which affect the nonfunctional requirements) that occur during the CPS's operations and affect the CPS's normal behavior. Therefore, they proposed an approach that can predict possible failures and ensure specific actions are taken for self-adapting the CPS to the new prevailing conditions.

The approach is based on the runtime-efficient probabilistic model-checking paradigm [67] and it uses DTMCs for the CPS's model description and PCTL to describe the requirements. The proposed solution requires executing two steps, one at design time (precomputation) and the other at runtime (verification). Once we execute the two steps, it results in a set of verification conditions that need evaluation as soon as changes occur, while also checking whether the system's requirements (reliability, performance, and power consumption) have been met. The proposed approach also enables sensitivity analysis at runtime, to understand the impact of changes of system property constituents on their values and to determine the CPS's adaptation strategies. Several experiments to assess the proposed approach were performed, and the results obtained showed significant improvements in terms of efficiency, as compared to model-checking algorithms that are computationally expensive and cannot be used at runtime.

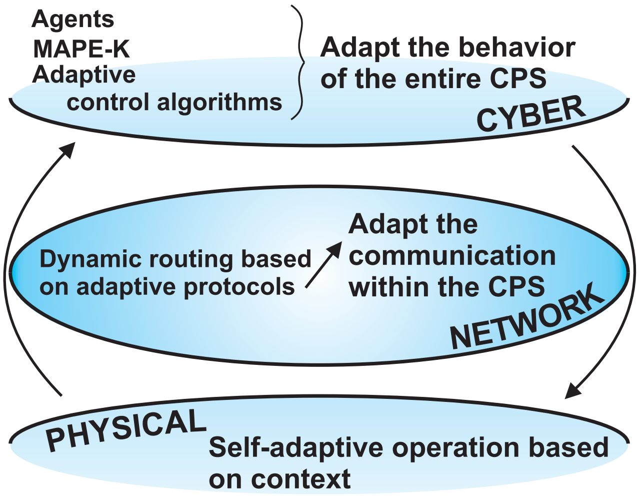
Agents
MAPE-K
Adaptive
control algorithms
Adapt the behavior of the entire CPS
CYBER
Dynamic routing based on adaptive protocols
Adapt the communication within the CPS
NETWORK
PHYSICAL
Self-adaptive operation based on context

FIGURE 3. Self-adaptive mechanisms in a CPS.

Table 1 summarizes the aforementioned self-adaptation approaches recently proposed for different types of CPSs, and highlights the application domain, self-adaptation techniques used, CPS architecture layers where these techniques are applied, and results obtained, along with their strengths and weaknesses. Since CPSs are software-intensive systems, most prior efforts related to self-adaptive mechanisms in CPSs have focused on actions performed at the cyber layer (see Table 1). Analysis of the literature also reveals that agents and MAS represent the most common ways to achieve self-adaptation mechanisms at CPSs' cyber layer, followed by formal models based on context. The design of self-adaptive mechanisms at the physical and middleware layers of CPSs have not really been explored by researchers, although some physical components and models have been developed for supporting the cyber part in system reconfiguration, in order for these components to properly achieve their tasks.

Based on our earlier discussion of related works on self-adaptation, in the next section we discuss the techniques needed to develop a decentralized and multilayered architecture for a CPS, which consists of cooperating components equipped with sensing and actuating, processing, storing, and networking capabilities.

IV. ARCHITECTURAL REQUIREMENT FOR SELF-ADAPTATION IN CPS

A general architecture of a CPS is based on three layers: the physical layer, composed of sensors, actuators and systems based on embedded processors; the networking layer, which facilitates communication between components in the CPS; and the cyber layer, which includes applications and services. Figure 3 depicts this architecture, with the self-adaptive mechanisms considered necessary for assuring the CPS's proper operation by taking into account different granularity levels, starting from the tiny physical sensor and actuating devices, and ending with the entire system of systems.

Adaptation should be implemented in all layers of a CPS, taking into consideration the tight coupling between them,

TABLE 1. Self-adaptation approaches in CPS research.

Research effort	Application domain	CPS architecture layer(s)	Results	Self-adaptation techniques	Strengths	Weaknesses and limitations
Architecture framework for experimentations with self-adaptive CPSs [46]	Transportation	Middleware	Formal self-adaptive model for CPSs Two runtime simulation frameworks (C++, Java)	Monitor-Analyze-Plan-Execute over shared Knowledge (MAPE-K) loops	Autonomous components and ensembles Decentralized adaptive approach Openness and extensibility for adaptation policies	Depends on network latencies and limited network connectivity
Parallel transportation management and control system and its applications in developing smart cities [49]	Transportation	Cyber	Artificial system built from the fusion of agent control, social signal, and the Internet of Things	Agents	Use of a parallel transportation management and control system, running alongside the real transportation system, allows adaptation based on hardware-in-the-loop simulations	Only the initial step in developing a fully autonomous self-adaptive system An operator is needed to choose the adaptation action
Building Energy Management Systems (BEMS) [50]	Smart buildings—smart cities	Physical	Thermal energy storage system with Computer Intelligence (CI)-based predictive control	CI techniques—ANNs, fuzzy logic, evolutionary algorithms	CI-based components suited for control tasks because of their ability to discover underlying interrelationships between data and learning specific control tasks based on such data	Validation only through a case study of CI-based control of a thermal energy storage unit
Smart grids: a CPS perspective [2]	Smart grids	Physical and cyber Micro-operational and macro-operational	Recommendation for a system of systems (or co-design) approach in the design and implementation of smart grids	Agents New generation of distributed control strategies	Implementation of distributed intelligence and optimization by using a multiagent system (MAS) to achieving self-adaptation, self-organization, and self-configuring capabilities Self-adaptation through distributed control	No industrial-grade agent-based platforms and integrated design models that can be applied Most decentralized control methods rely on system modeling with full states; this is infeasible for CPSs
Cloud-based software platform for big data analytics in smart grids [51]	Smart grids	Cyber	A cloud-based software platform for data-driven analytics that detects and pre-emptively corrects the supply-demand mismatch in smart grids	A cloud-based software system that can detect the supply-demand mismatch and pre-emptively correct it by initiating demand-side management	A semantic information integration pipeline, a secure repository, scalable machine-learning models, and a Web portal and mobile application	High costs for using commercial cloud providers Management of the hardware in private clouds could be a problem

to ensure the expected behavior of each CPS subsystem and their combined results. Furthermore, cross-layer approaches that enable information sharing across the physical, network, and cyber layers must be considered (such as component

TABLE 1. *(Continued.)* Self-adaptation approaches in CPS research.

Context-aware wireless mobile autonomic computing and communications: research trends and emerging applications [57]	Case study scenario based on a critical system for monitoring and rescue in case of hazard occurrence	Communication	Tolerance of significant active component losses with low impact on performance	Context-awareness applied to autonomic computing and to communication to enable self-adaptive mechanisms for modern software-intensive CPSs	Supports context-awareness and autonomic computing mechanisms	Validation through simulation Approach has not been demonstrated in a real environment
Current status and advancement of CPSs in manufacturing [58]	Manufacturing	Cyber and physical	An adaptive manufacturing CPS consisting of: –a multiagent architecture –commercial control boards capable of running the multiagent setup	Agents, Agent-oriented Architectures (AoA)	Uses an AoA approach Control demands are managed from an embedded system point of view, which means that each system's module is an entity with its own controls Physical components specially designed for an AoA approach	Mainly experimental systems that were not rigorously tested in real scenarios on the factory floor
Toward context-oriented self-adaptation in resource-constrained CPSs [63]	Resource-constrained applications	Cyber	Design for adaptability concepts: context and context group ConesC, a context-oriented extension for nesC to ensure adaptability of resource-constrained CPSs	Context-oriented software	Decoupled and cohesive context-oriented approach Software design can evolve easily over time	Preliminary solution evaluated only on a single test case Negligible runtime system overhead
Meta-adaptation strategies for adapting in CPSs [64]	Case study scenario based on a firefighter coordination system	Cyber	Two sets of tactics to adapt CPS, generated at runtime by knowledge exchange using data classification and process-period adjustment techniques	Meta-adaptation strategies	Higher potential than predesign strategies, in terms of expressive power and covering the problem space	Preliminary solution evaluated only on a single test case Requires dedicated hardware infrastructure and may affect other CPS resources (such as battery and network usage)
Modeling self-adaptive software systems with learning Petri nets [65]	Self-adaptive software systems	Cyber	Development of an adaptive Petri net (APN) for modeling self-adaptive software systems	APNs	Decision making based on runtime data and achieving software system-behavior models that can subsequently be used by model-checking tools	The approach is directed only to adaptive software systems; its application to CPSs is hampered by the interactions between the different composing

states, overall CPS state, components' predicted behavior based on performed actions, and others). As the related work discussions show, a CPS's design and development requires a significant amount of technical expertise and knowledge from

TABLE 1. ***(Continued.)*** **Self-adaptation approaches in CPS research.**

						layers (cyber, network, and physical)
Supporting self-adaptation via quantitative verification and sensitivity analysis at runtime [66]	Self-adaptive software systems	Cyber	Self-adaptive model for CPS that verifies whether systems' nonfunctional requirements are satisfied when environmental changes occur, and may invoke adaptation strategies at runtime	Runtime-efficient probabilistic model checking	Mathematical framework for runtime-efficient probabilistic model checking Substantial improvement in terms of efficiency compared to existing solutions	The model does not take into account the requirements' evolution and the interactions between the CPS's physical and networking components
Our proposed architecture for self-adaptation in CPSs	*All CPSs' application domains*	*Cyber, network, and physical*	*Self-adaptive CPSs*	*MAPE-K model* *Agents* *Ontology*	*Decentralized adaptive approach* *Manages the entire CPS to deal with uncertainties* *Takes into account the interactions between all the CPSs layers and the run-time management of requirements*	*Validation of real-life CPSs to occur in future work*

a broad range of research fields, including computational intelligence, control theory, communication, embedded systems, robust hardware design, and fault and error tolerance.

A. CYBER LAYER

The system's self-adaptation characteristic should be considered during the design phase, while also formalizing the entire system's operation. In this context, models and strategies for adapting the CPS behavior should be devised and incorporated in the CPS design's initial development steps. The MAPE-K model has proven useful in assuring a CPS's self-adaptability through its five main components: monitoring, analysis, planning, execution, and knowledge [36, 68]. This model should drive a CPS architecture and its first four elements (MAPE) should be implemented using techniques such as MASs, Petri nets, and Markov models. Agents' characteristics (autonomy, communication capabilities, reactivity, proactivity, and cooperation to achieve common goals) make them the best candidates for implementing decentralized and collaborative adaptive mechanisms at the cyber CPS layer. Furthermore, adaptive behavior is also a fundamental characteristic of intelligent agents [37], so MASs can easily implement MAPE components' primary functions. These functions include storing and filtering runtime data collected from sensors, analyzing collected data, developing a set of actions needed to adapt the system (e.g., reconfiguring the CPS structure, adjusting the CPS behavior during operation), and executing actions via the actuators. The MAPE components can be implemented using specialized agents that enact the aforementioned functions (e.g., acquisition, filtering, analysis, planning, and execution agents). The model's knowledge component must contain different system states' descriptions in every layer, for every physical device (e.g., sensors and actuators), throughout the environment, and of the adaptation policies; then agents use ontologies to implement them. Then agents—as intelligent computing entities—efficiently manage and work with the ontologies to support important CPS requirements such as interoperability, security, dependability, predictability, QoS, and sustainability. Knowledge component should assist other MAPE-K components (represented by agents) to achieve their goals. Figure 4 overviews a self-adaptive CPS with the MAPE-K model implemented using a MAS and ontology.

To summarize, self-adaptation should be employed in the cyber layer of a CPS for managing the entire system, to deal with uncertainties.

B. NETWORK LAYER

The network layer should also support self-adaptation capabilities for ensuring reliable and predictable communication between the components of the CPS and between the CPS and its environment. This layer should seamlessly adapt to different network loads and react to security threats and changes in the environment. There is an increasing trend in using

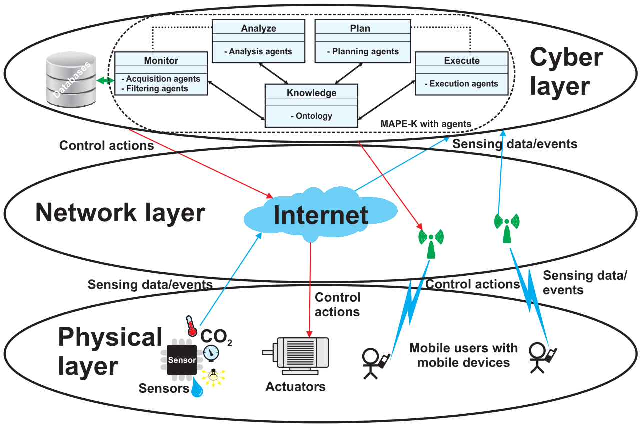

FIGURE 4. Overview of a self-adaptive CPS with the MAPE-K model implemented using a MAS and ontology.

wireless communication, which, because of the medium's transmission characteristics, is subject to major reliability issues. As a result, we need to continuously monitor networking issues (e.g., packet losses, delays, data corruption, network topology changes, and malicious attacks) that may occur during data transmissions and be able to adapt the communication protocol for assuring a satisfactory level of reliability, predictability, and dependability. Although several research efforts [69]–[72] have proposed implementing a wide range of Medium Access Control (MAC) protocols in the wireless domain, most of the approaches dealt with power consumption, and only a few of them [72]–[75] proposed mechanisms for assuring high reliability and availability levels, while maintaining high throughput [76]. The devices communicating within the CPS should be provided with self-diagnosis capabilities, so that they can switch between different options (such as the protocol being used for data transmission, the rate at which transmission and reception actions are performed) for transmitting and receiving network data. In other words, CPS components must be capable of adapting to context changes (e.g., selecting the best standard for data transmission, choosing between Wi-Fi, Zigbee, Bluetooth, or others depending on the availability of relaying nodes or gateways). To summarize, self-adaptation should be employed in this layer to achieve communication resilience, namely the network ability to support and maintain an acceptable level of service despite the occurrence of faults and other factors that affect normal operations [77].

C. PHYSICAL LAYER

The physical layer is also subject to fault and error occurrences, and techniques for mitigating them to continue operating the CPS without significantly degrading performance and QoS should be ensured. In particular, the physical devices should adapt their operations based on context-monitoring results and the self-assessment of their current state. To achieve these objectives, we must implement fault-tolerant schemes in the hardware or software that runs on various heterogeneous devices that operate in the CPS's physical layer. However, usually the digital systems within CPSs are constrained in terms of physical and computational resources. Despite these design constraints, we must develop efficient mechanisms for providing CPS subsystems with self-monitoring capabilities, to prevent or correct abnormal behavior on the fly. We need adaptation to respond to fault and error occurrences, as well as changing environmental conditions or the internal CPS's state. For example, a wireless node reporting environmental condition parameters probably has to adapt its way of operation, depending on its location and level of power supply. So, we must adapt the power-of-transmission policy according to the occurrence of conditions and factors that may cause interference or the distance from the communication partner. At the same time, the CPS device's communication rate may require adjustment, depending on its power-supply level. To achieve the aforementioned goals, we need a thorough analysis of the operating context, starting from the design phase, and to implement mechanisms that can react to unpredictable changes, leading to timely system adaptation.

As we mentioned earlier, developing and implementing self-adaptation mechanisms is a fundamental requirement in CPS architecture design, to satisfy one of their most important characteristics: dependability. However, the current literature reveals that research in this area of CPS is still in its infancy, with most systems validated mainly by using simple case studies. More research is needed, to develop novel CPS mechanisms that adequately contend with the uncertainties caused by implementing feedback loops for adaptation at different levels in CPSs. Although we need coarse- and fine-grained self-adaptive mechanisms [78, 79], the level of interaction between different loops within CPSs (feedback or nested control loops) require further study.

V. CONCLUSION

CPSs are vastly engineered systems that require an innovative perspective in their design and development, to anticipate completely new characteristics on a grander scale than ever previously encountered. To fully fathom the challenges and opportunities ahead, we must adopt a holistic view of CPSs that includes self-adaptation, autonomy, efficiency, functionality, reliability, safety, scalability, and usability. Here, we focused on CPSs' self-adaptation with a review of state-of-the-art self-adaptive approaches recently proposed by researchers for large-scale, complex systems. We further identified the approaches' strengths and weaknesses. Although a wealth of research exists on self-adaptation mechanisms in ICT systems, most previous efforts focused on developing software to ensure self-adaptation but neglected the implementation of self-adaptation for hardware components. Only a few results focused on CPSs, where a strong interaction occurs between the cyber and physical parts. Based on our literature review for self-adaptation techniques in CPSs, we discussed the techniques needed to build a general CPS architecture that provides efficient self-adaptation characteristics. Current work shows that research in this area is in its nascence, with further research opportunities abounding—including the development of cost-effective self-adaptation cross-layer solutions, as well as runtime model-driven approaches that manage requirements.

REFERENCES

[1] National Science Foundation (NSF), Arlington, VA, USA. (2013). *Cyber Physical Systems NSF10515*. Accessed Oct. 2019. [Online]. Available: http://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm

[2] X. H. Yu and Y. S. Xue, "Smart grids: A cyber-physical systems perspective," *Proc. IEEE*, vol. 104, no. 5, pp. 1058–1070, May 2016.

[3] L. Ribeiro and M. Björkman, "Transitioning from standard automation solutions to cyber-physical production systems: An assessment of critical conceptual and technical challenges," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3816–3827, Dec. 2018.

[4] M. A. Rahman, S. Azad, A. T. Asyhari, M. Z. A. Bhuiyan, and K. Anwar, "Collab-SAR: A collaborative avalanche search-and-rescue missions exploiting hostile alpine networks," *IEEE Access*, vol. 6, pp. 42094–42107, 2018.

[5] M. A. Rahman, A. T. Asyhari, S. Azad, M. M. Hasan, C. P. C. Munaiseche, and M. Krisnanda, "A cyber-enabled mission-critical system for post-flood response: Exploiting TV white space as network backhaul links," *IEEE Access*, vol. 7, pp. 100318–100331, 2019.

[6] R. S. Baheti and H. Gill, *The Impact of Control Technology*, T. Samad and A. M. Annaswamy, Eds. New York, NY, USA: IEEE Control Systems Society, 2011, pp. 161–166. Accessed: Oct. 2019. [Online]. Available: https://s3-us-west-2.amazonaws.com/cc-ieeecss/IoCT-FullReport_v2.pdf

[7] I. Horváth and B. H. M. Gerritsen, "Cyber-physical systems: Concepts, technologies and implementation principles," in *Proc. TMCE*, vol. 5, May 2012, pp. 19–36.

[8] M. Conti, S. K. Das, C. Bisdikian, M. Kumar, L. M. Ni, A. Passarella, G. Roussos, G. Tröster, G. Tsudik, and F. Zambonelli, "Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence," *Pervasive Mobile Comput.*, vol. 8, no. 1, pp. 2–21, Feb. 2012.

[9] B. X. Huang, "Cyber physical systems: A survey," Presentation Rep., Jun. 2008.

[10] Steering Committee for Foundations in Innovation for Cyber-Physical Systems. (2013). *Foundations for Innovation: Strategic R&D Opportunities for 21st Century Cyber-Physical Systems*. Accessed: Oct. 2019. [Online]. Available: http://www.nist.gov/el/upload/12-Cyber-Physical-Systems020113_final.pdf

[11] J. Sztipanovits, T. Bapty, X. Koutsoukos, Z. Lattmann, S. Neema, and E. Jackson, "Model and tool integration platforms for cyber–physical system design," *Proc. IEEE*, vol. 106, no. 9, pp. 1501–1526, Sep. 2018.

[12] X. Lyu, Y. Ding, and S.-H. Yang, "Safety and security risk assessment in cyber-physical systems," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 4, no. 3, pp. 221–232, Sep. 2019.

[13] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci, "A roadmap toward the resilient Internet of Things for cyber-physical systems," *IEEE Access*, vol. 7, pp. 13260–13283, 2019.

[14] T. Shah, A. Yavari, K. Mitra, S. Saguna, P. P. Jayaraman, F. Rabhi, and R. Ranjan, "Remote health care cyber-physical system: Quality of service (QoS) challenges and opportunities," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 1, no. 1, pp. 40–48, 2016.

[15] S. Kim, Y. Won, I.-H. Park, Y. Eun, and K.-J. Park, "Cyber-physical vulnerability analysis of communication-based train control," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6353–6362, Aug. 2019.

[16] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A survey on concepts, applications, and challenges in cyber-physical systems," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 12, pp. 1–27, Dec. 2014.

[17] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *Proc. IEEE Int. Conf. Wireless Commun. Signal Process.*, 2011, pp. 1–6.

[18] S. K. Khaitan and J. D. McCalley, "Design techniques and applications of cyberphysical systems: A survey," *IEEE Syst. J.*, vol. 9, no. 2, pp. 350–365, Jun. 2015.

[19] J. K. S. Lau, C.-K. Tham, and T. Luo, "Participatory cyber physical system in public transport application," in *Proc. IEEE Int. Conf. Utility Cloud Comput.*, Dec. 2011, pp. 355–360.

[20] H. Yan, J. F. Wan, and H. Suo, "Adaptive resource management for cyber-physical systems," *Appl. Mech. Mater.*, vols. 157–158, pp. 747–751, Feb. 2012.

[21] S. Lim, L. Chung, O. Han, and J.-H. Kim, "An interactive cyber-physical system (CPS) for people with disability and frail elderly people," in *Proc. 5th Int. Conf. Ubiquitous Inf. Manage. Commun. (ICUIMC)*, Feb. 2011, Art. no. 113.

[22] Z.-Y. Bai and X.-Y. Huang, "Design and implementation of a cyber physical system for building smart living spaces," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 5, pp. 764186-1–764186-9, 2012.

[23] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 263–284, 1st Quart., 2016.

[24] A. K. Tyagi, "Cyber physical systems (CPSs)—Opportunities and challenges for improving cyber security," *Int. J. Comput. Appl.*, vol. 137, no. 14, pp. 19–27, Mar. 2016.

[25] P. Marwedel and M. Engel, "Cyber-physical systems: Opportunities, challenges and (some) solutions," in *Management of Cyber Physical Objects in the Future Internet of Things* (Internet of Things), A. Guerrieri, V. Loscri, A. Rovella, and G. Fortino, Eds. Cham, Switzerland: Springer, 2016.

[26] H. Muccini, M. Sharaf, and D. Weyns, "Self-adaptation for cyber-physical systems: A systematic literature review," in *Proc. 11th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst. (SEAMS)*, New York, NY, USA, 2016, pp. 75–81.

[27] T. Lints, "The essentials of defining adaptation," in *Proc. 4th IEEE Int. Syst. Conf.*, San Diego, CA, USA, Apr. 2010, pp. 113–116.

[28] M. Levesque, M. Maier, Y. Desai, and G. Joos, "Adaptive admission control for a smart grid FiWi communications network facing power blackouts during a DDoS attack," in *Proc. IEEE Green Technol. Conf.*, Tulsa, OK, USA, Apr. 2012, pp. 1–3.

[29] D. Garlan, B. Schmerl, and S.-W. Cheng, "Software architecture-based self-adaptation," in *Autonomic Computing and Networking*, Y. Zhang, L. Yang, and M. Denko, Eds. Boston, MA, USA: Springer, 2009.

[30] S. Yatsyshyn and B. Stadnyk, *Cyber-Physical Systems: Metrological Issues*, 1st ed. Barcelona, Spain: IFSA Publishing, 2017.

[31] R. Laddaga, "Self-adaptive software," DARPA BAA, Arlington County, VA, USA, Tech. Rep. 98-12, 1997.

[32] D. Blaauw and S. Das, "CPU, heal thyself," *IEEE Spectr.*, vol. 46, no. 8, pp. 40–56, Aug. 2009.

[33] M.-I. Neagu, G. D. Moiş, and L. C. Miclea, "On-line error detection for tuning dynamic frequency scaling," in *Proc. IEEE Int. Conf. Automat., Qual. Test., Robot.*, Cluj-Napoca, Romania, May 2012, pp. 290–295.

[34] S. Wiesner, C. Gorldt, M. Soeken, K.-D. Thoben, and R. Drechsler, "Requirements engineering for cyber-physical systems," in *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World.* Berlin, Germany: Springer, Sep. 2014, pp. 281–288.

[35] A. Qasim and S. A. R. Kazmi, "MAPE-K interfaces for formal modeling of real-time self-adaptive multi-agent systems," *IEEE Access*, vol. 4, pp. 4946–4958, 2016.

[36] P. Arcaini, E. Riccobene, and P. Scandurra, "Modeling and analyzing MAPE-K feedback loops for self-adaptation," in *Proc. IEEE/ACM 10th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst.*, Florence, Italy, May 2015, pp. 13–23.

[37] T. Sanislav, S. Zeadally, and G. D. Mois, "A cloud-integrated, multilayered, agent-based cyber-physical system architecture," *Computer*, vol. 50, no. 4, pp. 27–37, Apr. 2017.

[38] T. Sanislav, S. Zeadally, G. Mois, and H. Fouchal, "Multi-agent architecture for reliable cyber-physical systems (CPS)," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Heraklion, Greece, Jul. 2017, pp. 170–175.

[39] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber–physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1086–1101, May 2016.

[40] Y. Zhang, C. Qian, J. Lv, and Y. Liu, "Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 737–747, Apr. 2017.

[41] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution," *Comput. Ind.*, vol. 66, pp. 99–111, Jan. 2015.

[42] Y. Brun, G. Di Marzo Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Müller, M. Pezzè, and M. Shaw, "Engineering self-adaptive systems through feedback loops," in *Software Engineering for Self-Adaptive Systems* (Lecture Notes in Computer Science), vol. 5525, B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, Eds. Berlin, Germany: Springer-Verlag, 2009, pp. 48–70.

[43] G. A. Dumont and M. Huzmezan, "Concepts, methods and techniques in adaptive control," in *Proc. Amer. Control Conf.*, Anchorage, AK, USA, vol. 2, May 2002, pp. 1137–1150.

[44] J. H. Lumkes, *Control Strategies for Dynamic Systems: Design and Implementation*. Boca Raton, FL, USA: CRC Press, 2001, p. 612.

[45] R. de Lemos, D. Garlan, C. Ghezzi, H. Giese, J. Andersson, M. Litoiu, B. Schmerl, D. Weyns, L. Baresi, N. Bencomo, Y. Brun, J. Camara, R. Calinescu, M. B. Cohen, and A. Gorla, "Software engineering for self-adaptive systems: Research challenges in the provision of assurances," in *Software Engineering for Self-Adaptive Systems III. Assurances* (Lecture Notes in Computer Science), vol. 9640, R. de Lemos, D. Garlan, C. Ghezzi, and H. Giese, Eds. Cham, Switzerland: Springer, 2017.

[46] M. Kit, I. Gerostathopoulos, T. Bures, P. Hnetynka, and F. Plasil, "An architecture framework for experimentations with self-adaptive cyber-physical systems," in *Proc. IEEE/ACM 10th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst.*, Florence, Italy, May 2015, pp. 93–96.

[47] I. K. J. Gerostathopoulos, T. Bures, M. Kit, and F. Plasil, "Software engineering for software-intensive cyber-physical systems," in *Informatik*, E. Plödereder, L. Grunske, E. Schneider, and D. Ull, Eds. Bonn, Germany: Gesellschaft für Informatik, 2014, pp. 1179–1190.

[48] G. Blair, N. Bencomo, and R. B. France, "Models@ run.time," *Computer*, vol. 42, no. 10, pp. 22–27, Oct. 2009.

[49] F. Zhu, Z. Li, S. Chen, and G. Xiong, "Parallel transportation management and control system and its applications in building smart cities," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1576–1585, Jun. 2016.

[50] M. Manic, D. Wijayasekara, K. Amarasinghe, and J. J. Rodriguez-Andina, "Building energy management systems: The age of intelligent and adaptive buildings," *IEEE Ind. Electron. Mag.*, vol. 10, no. 1, pp. 25–39, Mar. 2016.

[51] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna, "Cloud-based software platform for big data analytics in smart grids," *Comput. Sci. Eng.*, vol. 15, no. 4, pp. 38–47, Jul./Aug. 2013.

[52] M. H. Athari and Z. Wang, "Impacts of wind power uncertainty on grid vulnerability to cascading overload failures," *IEEE Trans. Sustain. Energy*, vol. 9, no. 1, pp. 128–137, Jan. 2018.

[53] A. Srivastava, T. Morris, T. Ernster, C. Vellaithurai, S. Pan, and U. Adhikari, "Modeling cyber-physical vulnerability of the smart grid with incomplete information," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 235–244, Mar. 2013.

[54] A. A. Babalola, R. Belkacemi, and S. Zarrabian, "Real-time cascading failures prevention for multiple contingencies in smart grids through a multi-agent system," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 373–385, Jan. 2018.

[55] M. Negnevitsky, N. Voropai, V. Kurbatsky, N. Tomin, and D. Panasetsky, "Development of an intelligent system for preventing large-scale emergencies in power systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Vancouver, BC, Canada, Jul. 2013, pp. 1–5.

[56] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: Pricing methods and optimization algorithms," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 152–178, Mar. 2015.

[57] F. Chiti, R. Fantacci, M. Loreti, and R. Pugliese, "Context-aware wireless mobile autonomic computing and communications: Research trends and emerging applications," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 86–92, Apr. 2016.

[58] L. Wang, M. Törngren, and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," *J. Manuf. Syst.*, vol. 37, pp. 517–527, Oct. 2015.

[59] P. Leitao, V. Marik, and P. Vrba, "Past, present, and future of industrial agent applications," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2360–2372, Nov. 2013.

[60] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: PROSA," *Comput. Ind.*, vol. 37, no. 3, pp. 255–274, Nov. 1998.

[61] P. Neves and J. Barata, "Evolvable production systems," in *Proc. IEEE Int. Symp. Assem. Manuf. (ISAM)*, Suwon, South Korea, Nov. 2009, pp. 189–195.

[62] V. Botti and A. Giret, "ANEMONA development process," in *ANEMONA* (Advanced Manufacturing). London, U.K.: Springer, 2008, pp. 91–133.

[63] M. Afanasov, L. Mottola, and C. Ghezzi, "Towards context-oriented self-adaptation in resource-constrained cyberphysical systems," in *Proc. IEEE 38th Int. Comput. Softw. Appl. Conf. Workshops*, Vasteras, Sweden, Jul. 2014, pp. 372–377.

[64] I. Gerostathopoulos, T. Bures, P. Hnetynka, A. Hujecek, F. Plasil, and D. Skoda, "Meta-adaptation strategies for adaptation in cyber-physical systems," in *Proc. 9th Eur. Conf. Softw. Archit. (ECSA)*, D. Weyns, R. Mirandola, and I. Crnkovic, Eds. Cavtat, Croatia: Springer, Sep. 2015, pp. 45–52.

[65] Z. Ding, Y. Zhou, and M. Zhou, "Modeling self-adaptive software systems with learning Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 4, pp. 483–498, Apr. 2016.

[66] A. Filieri, G. Tamburrelli, and C. Ghezzi, "Supporting self-adaptation via quantitative verification and sensitivity analysis at run time," *IEEE Trans. Softw. Eng.*, vol. 42, no. 1, pp. 75–99, Jan. 2016.

[67] A. Filieri, C. Ghezzi, and G. Tamburrelli, "Run-time efficient probabilistic model checking," in *Proc. IEEE 33rd Int. Conf. Softw. Eng.*, May 2011, pp. 341–350.

[68] I. Elgendi, M. F. Hossain, A. Jamalipour, and K. S. Munasinghe, "Protecting cyber physical systems using a learned MAPE-K model," *IEEE Access*, vol. 7, pp. 90954–90963, 2019.

[69] M. Zareei, A. Taghizadeh, R. Budiarto, and T.-C. Wan, "EMS-MAC: Energy efficient contention-based medium access control protocol for mobile sensor networks," *Comput. J.*, vol. 54, no. 12, pp. 1963–1972, Nov. 2011.

[70] C. Cano, B. Bellalta, A. Sfairopoulou, and J. Barcelo, "A low power listening MAC with scheduled wake up after transmissions for WSNs," *IEEE Commun. Lett.*, vol. 13, no. 4, pp. 221–223, Apr. 2009.

[71] T. Ha, J. Kim, and J.-M. Chung, "HE-MAC: Harvest-then-transmit based modified EDCF MAC protocol for wireless powered sensor networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 3–16, Jan. 2018.

[72] S. Pudasaini, S. Shin, and K. Kim, "Throughput and reliability analysis of a scalable broadcast MAC for distributed wireless networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, Aug. 2012, Art. no. 254.

[73] D. N. M. Dang, C. S. Hong, S. Lee, and E.-N. Huh, "An efficient and reliable MAC in VANETs," *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 616–619, Apr. 2014.

[74] S. Moulik, S. Misra, and D. Das, "AT-MAC: Adaptive MAC-frame payload tuning for reliable communication in wireless body area networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 6, pp. 1516–1529, Jun. 2017.

[75] S. Misra, P. V. Krishna, V. Saritha, H. Agarwal, L. Shu, and M. S. Obaidat, "Efficient medium access control for cyber–physical systems with heterogeneous networks," *IEEE Syst. J.*, vol. 9, no. 1, pp. 22–30, Mar. 2015.

[76] T. Sanislav, G. Mois, S. Folea, and L. Miclea, "Integrating wireless sensor networks and cyber-physical systems: Challenges and opportunities," in *Cyber-Physical System Design With Sensor Networking Technologies* (Control, Robotics and Sensors). London, U.K.: IET Digital Library, 2016, ch. 3, pp. 47–76. [Online]. Available: http://digital-library.theiet.org/content/books/10.1049/pbce096e_ch3, doi: 10.1049/PBCE096E_ch3.

[77] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, pp. 1245–1265, Jun. 2010.

[78] S. Hassan and R. Bahsoon, "Microservices and their design trade-offs: A self-adaptive roadmap," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, San Francisco, CA, USA, Jun./Jul. 2016, pp. 813–818.

[79] G. W. Greenwood and A. M. Tyrrell, *Introduction to Evolvable Hardware: A Practical Guide for Designing Self-Adaptive Systems*. Hoboken, NJ, USA: Wiley, 2006, p. 224.

SHERALI ZEADALLY earned his bachelor's degree in computer science from the University of Cambridge, England. He also received a doctoral degree in computer science from the University of Buckingham, England. He is currently an Associate Professor in the College of Communication and Information, University of Kentucky. His research interests include Cybersecurity, privacy, Internet of Things, computer networks, and energy-efficient networking. He is a Fellow of the British Computer Society and the Institution of Engineering Technology, England.

TEODORA SANISLAV (M'12) received the B.S. degree in computer science and the Ph.D. degree in systems engineering from the Technical University of Cluj-Napoca, Romania, in 2003 and 2013, respectively. From 2004 to 2011, she was a Scientific Researcher in a research institute in the field of automation. She is currently a Lecturer with the Automation Department, Technical University of Cluj-Napoca. Her current research interests include cyber-physical systems, dependability analysis, and intelligent systems.

GEORGE DAN MOIS (M'08) received the Ph.D. degree in systems engineering from the Technical University of Cluj-Napoca, Cluj-Napoca, Romania, in 2011. He is currently a Lecturer with the Automation Department, Technical University of Cluj-Napoca. Since 2017, he has been a part of the Innovation and Universities Group, Bosch Engineering Center Cluj. His current research interests include embedded system design and wireless sensor networks.