



## MIDDLEWARE CHALLENGES FOR CYBER-PHYSICAL SYSTEMS

NADER MOHAMED<sup>\*</sup>, JAMEELA AL-JAROODI<sup>†</sup>, SANJA LAZAROVA-MOLNAR<sup>‡</sup> AND IMAD JAWHAR<sup>§</sup>

**Abstract.** Cyber-Physical Systems (CPS) are being developed to provide useful interactions between physical systems and environments and cyber world for a variety of applications. CPS are designed with a set of software and interconnected distributed hardware components that are linked with physical elements to provide advanced monitoring and control mechanisms geared towards enhancing the targeted physical system or environment. These components function seamlessly to offer specific functionalities that help enhance human lives, physical system operations and environments. While CPS can offer many smart enhancements for improving physical processes, the development of such complex systems composed of many distributed and heterogeneous components is extremely difficult. This is due to the many communication, computing, and networking challenges. Using an appropriate middleware that provides a framework to support developing and operating diverse CPS applications is a novel method to address these challenges. The availability of advanced middleware services and platforms can provide effective approaches for enhancing CPS application development processes as well as provide more robust environments for operating CPS applications. Such middleware can significantly reduce the time needed to design, build, test, and operate robust CPS applications. However, designing a common middleware platform for diverse types of CPS applications is not trivial. This paper investigates the middleware challenges for CPS, based on the different types of CPS applications being developed and their specific challenges. In addition, the paper discusses the current efforts of developing middleware platforms for CPS and the open research issues in the field.

**Key words:** Cyber-physical systems, Middleware, Software engineering, Application development.

**AMS subject classifications.** 68U99

**1. Introduction.** There are many CPS applications that add enhancements and smart features to several types of physical systems and environments. CPS can add smart mechanisms to fully automate manufacturing processes, manage and enhance the operations and safety of environments and infrastructures, and enable Unmanned Autonomous Vehicle (UAV) operations and applications. They can also enhance the safety of transportation systems, enhance energy consumption in smart buildings, and improve healthcare for patients. CPS combine various concepts and technologies from embedded systems, networks, distributed systems, software, and hardware; as well as other engineering disciplines such as systems, mechanical, control, civil, and electrical engineering to provide added features to the physical world [67].

While CPS can offer many smart enhancements for improving physical systems and processes, the development of such complex systems composed of many components interacting in various ways and capabilities is extremely difficult. CPS attach different hardware components like sensors, actuators, microcontrollers, and other devices to physical systems or environments and use distributed software that implements smart algorithms to control the corresponding physical system [48, 35]. The distribution and heterogeneity of these devices and their links with the physical components make the design, development, and operations of CPS very challenging.

Due to the importance of CPS applications and the complexity of their development process, huge research efforts started investigating the different issues associated with CPS and their applications. These include security, reliability, performance, quality, validation, and development methodologies and tools [48]. In our previous work, we discussed the importance of software components in any CPS and highlighted the main software engineering issues that include the complexity of analysis, design, development, and testing for CPS software [20]. We also briefly highlighted different research directions to tackle these issues. One of these directions is to develop and use advanced middleware platforms to support CPS applications. In this paper, we investigate in detail the middleware challenges for CPS. This investigation is based on the different types of CPS applications and their specific challenges. This paper can help CPS applications developers to recognize the middleware challenges and requirements of different CPS applications. As a result, suitable approaches can be selected and adapted to fit the specific needs of the applications being considered. Current and new approaches

<sup>\*</sup>Middleware Technologies Lab., Pittsburgh, Pennsylvania, USA ([nader@middleware-tech.net](mailto:nader@middleware-tech.net)).

<sup>†</sup>Department of Engineering, Robert Morris University, Moon Township, Pennsylvania, USA ([aljaroodi@rmu.edu](mailto:aljaroodi@rmu.edu)).

<sup>‡</sup>University of Southern Denmark, Odense, Denmark ([slmo@mmmi.sdu.dk](mailto:slmo@mmmi.sdu.dk)).

<sup>§</sup>Midcomp Research Center, Saida, Lebanon ([imad@midcomp.net](mailto:imad@midcomp.net)).

for middleware platforms need to be considered, yet most of the current approaches may not fit well with the nature of CPS applications. For example, a middleware platform such as real-time CORBA has shown promise for general distributed systems with some constraints; however, it lacks the flexibility in dealing with the CPS challenges.

Middleware challenges for related systems such as Wireless Sensor Networks (WSNs) and the Internet of Things (IoT) were thoroughly investigated [70, 60, 68, 25]. Furthermore, several papers investigated and discussed general challenges in CPS [48, 35, 74]. However, none investigated in detail the middleware challenges for CPS. Identifying and studying middleware challenges is important to be able to design and develop the most suitable middleware platform to support different CPS applications. In our previous work [58], we briefly highlighted how middleware can provide different types of support for CPS. In this paper, we extend our previous work including detailed middleware challenges, current research efforts in this regard, and discuss several open research issues that need to be investigated and resolved to offer the best possible middleware platforms for CPS.

In the rest of the paper, we offer some background about CPS and middleware in Section 2. In Section 3, we discuss different CPS applications to understand the role middleware can play to support them. We then discuss and identify the general and common challenges of CPS applications in Section 4. Section 5 discusses the role of middleware to support CPS applications while middleware challenges for CPS are identified and discussed in Section 6. The current research efforts for CPS middleware are discussed in Section 7 and Section 8 lists some open research issues in the field. Finally, we offer some concluding remarks and future directions in Section 9.

**2. Background.** Here we provide preliminary information about CPS and middleware. This establishes a base line for the upcoming discussion of middleware platforms for CPS.

**2.1. Cyber-Physical Systems.** CPS are distributed embedded systems developed to support smart and context-aware mission-critical applications in different domains such as energy, manufacturing, healthcare, civil infrastructures, automotive, transportation, aerospace, entertainment, and consumer appliances. CPS provide monitoring and control functions to help achieve specified goals to benefit the application domain. Unlike traditional embedded systems, CPS are complex embedded systems with distributed components and processing capabilities. Embedded computing devices of CPS can be in sensors, actuators, microcontrollers, and other devices usually connected with a wired or wireless network and tightly coupled with their physical environment.

As CPS are embedded in physical environments, they provide useful interactions between the computational and physical elements through intelligent mechanisms. These intelligent mechanisms can be organized in four main steps:

- Observing the status of the physical system or environment using different type of sensors that are attached to the elements of the physical system or environment.
- Building a knowledgebase about the physical system environment from the collected sensed information using software functions and storage systems.
- Making decisions to enhance or control the physical system or environment to meet specific objectives defined for the application. This is done using the knowledgebase and some status information of the physical system or environment with the help of smart algorithms that run on some integrated computing components.
- Applying the enhancement or control actions using actuators that are attached to the elements of the physical system or environment.

These four steps are linked in a closed loop as shown in Figure 2.1 to allow the CPS to provide full monitoring and control functions to achieve the needed objectives. These objectives can be to provide the adaptability, autonomy, efficiency, functionality, reliability, safety, and usability of the system or environment.

**2.2. Middleware.** Middleware for a distributed system is a logical software layer that abstracts the details of the underlying distributed components and provides a set of services to develop and operate distributed software applications beyond those available from the individual components. The provided services can be used to support integration, system management, runtime for distributed code, fault tolerance, security, or



FIG. 2.1. Closed-loop control steps of CPS

load-balancing. In addition, it can provide more advanced services to support issues related to power consumption, limited resources devices, etc.

Middleware has become a necessary part of distributed computing. It is practically impossible to develop large-scale distributed systems or applications without involving middleware [18]. By comparison, trying to build a distributed systems or application without middleware is like trying to write a simple application on a personal computer without the operating system. Generally, the main functions of different middleware platforms are [18]:

- Providing tools to simplify the development of complex distributed applications.
- Providing high-level abstractions and interfaces to facilitate distributed application integration and reuse.
- Hiding the heterogeneity of the underlying environments.
- Hiding the distribution and communication details in the underlying environment.
- Enabling communications among the different distributed components of the infrastructure.
- Offering services for common functions needed by different applications to reduce development and duplication efforts.
- Providing a common architecture to enable adding new services and features without having to change the distributed applications.
- Providing added-value features and nonfunctional properties such as security, reliability, scalability, and Quality of Services (QoS).

Due to the advantages of using middleware, several middleware platforms were developed for different distributed applications and environments. Examples include WSNs [66], cloud computing [78], collaborative UAVs [57], and mobile social networks [38]. The developed middleware solved many issues in these environments and similarly can solve many issues in CPS.

**3. CPS Applications.** As CPS can provide useful interactions between physical and cyber worlds, a number of domains such as health, energy, transportation, and security can greatly benefit from CPS applications [35]. However, developing and operating such applications can face with many challenges. To identify and understand these challenges, we discuss some important CPS applications used or proposed for different application domains. We highlight their benefits as well as their development and operations challenges. This will help us identify the type of support needed by the middleware platforms designed for CPS applications.

In the energy domain, CPS are used to add values such as efficiency, reliability, and sustainability of the production and distribution of electric power in smart grids [43]. A smart grid is a renovated electrical grid system that uses information and communication technology (ICT) to collect and act on available information on the behaviors of suppliers and consumers in an automated fashion. A smart grid is a CPS that provides self-monitoring and advanced control mechanisms for power production and distribution, as well as addressing consumer needs towards increased grid efficiency and reliability. This involves placing smart sensors and meters on production, transmission, and distribution systems in addition to consumers locations to get granular near real-time data about the current power production, consumption, and faults. Although the smart grid has many potential benefits, it requires the collection and analysis of huge amounts of data continuously. This collected data is processed in real-time to send back control information to adjust the operational conditions and improve efficiency, reliability, economics, and sustainability of the system. In addition, the processes

used to generate renewable energy from hydropower plants [56] and wind power plants [75] are controlled by CPS. Furthermore, the energy consumption in smart buildings is controlled and monitored by CPS [36]. The buildings equipment such as HVAC (Heating, Ventilating, and Air-Conditioning) systems, appliances, and lighting systems are controlled with CPS. Smart buildings CPS are usually equipped with different types of sensor nodes that monitor the current energy usage and environmental conditions. These sensors report their observations and measurements to a centralized CPS monitoring and control system. The control system implements intelligent algorithms to control the systems used in the buildings to optimize energy use based on the sensed observations and current operational and environmental conditions. CPS also provide control mechanisms for energy efficiency in data centers [79, 62].

In the health domain, many medical systems are controlled by CPS. Medical CPS should provide safe and intelligent continuous care for patients [49]. Medical CPS are networks of medical sensor devices that provide clinical monitoring functions such as heart-rate and oxygen levels; medical delivery devices such as infusion pumps and ventilators; and control devices that provide the main controls for medical CPS and are responsible for efficient and safe operations of the whole system. As medical CPS are life-critical systems, they must be context-aware, reliable and resilient to faults. Some medical CPS can be wearable systems and continuously used by patients. These wearable systems usually operate on battery power. In this case the medical CPS must be energy-efficient and designed to minimize energy consumption to extend the life of the devices.

In the transportation domain, an important CPS application area that recently received high attention is the vehicular safety applications. There are many safety applications for vehicles including lane change warning messages, emergency braking, collision avoidance mechanisms, and blind spot monitoring. These applications provide fully automatic or semi-automatic actions to enhance driving safety. Some of these applications are based on individual vehicle observations, decisions, and actions while others are based on collaborative observations, decisions, and actions where neighboring vehicles exchange messages for that purpose [53]. The various devices and components needed to achieve these functionalities form the vehicular CPS [30]. Embedded software is used to implement different safety applications. The most important features of vehicular safety applications are the real-time and reliability support in detection and response. All aspects of vehicular safety applications including threat observations, decision making, communication, and actions must be in real-time and reliable. This imposes a serious restriction on how the software is designed and how well it supports high levels of integration across all the devices involved to ensure real-time and reliable responses. In addition, self-driving cars are considered as CPS [24]. Since they practically integrate all the mentioned features in addition to vision and monitoring components to allow the vehicle to navigate the roads based on sensed data and intelligent software that interprets and responds to this data in real-time. Other transport CPS include intelligent traffic light controls which include monitoring devices across multiple locations to accurately predict traffic patterns and adjust traffic lights to optimize flow. One example of such domain is discussed in [71].

In the security domain, CPS can be used to monitor and protect important large-scale infrastructures such as long oil and gas pipelines that extend for hundreds to thousands of miles [22]. These usually extend across unattended and sometimes difficult areas such as underwater, deserts and forests. Pipelines are considered important infrastructures that need to be monitored and protected as they provide the main supplies of energy and water for many countries and areas. Sensors, actuators, and other devices are usually deployed and connected using wired or wireless networks to monitor, control, and protect such infrastructures. In addition, CPS can be used to protect water networks and to make them smarter, more efficient, more reliable, and more sustainable. CPS can be embedded within water networks to involve some monitoring and control mechanisms and to add smart features to the operations of water distribution [44]. One of these functions is to provide early warning mechanisms to identify problems in water networks. For examples leaks and pipe bursts can be easily detected while fast and temporary solutions can be applied to reduce water waste and to minimize further risks or damages to the network.

Other CPS applications include greenhouses efficient control that aims to provide efficient control for suitable climate, soil, lighting, and water level in greenhouses [32]. Similarly, they can be used to enable smart homes [41]. In addition, CPS are used to autonomously operate unmanned vehicles CPS provide networks that connect the payloads on these vehicles like sensors, actuators, cameras, storage, communication devices, and the microcontrollers of the vehicle [57]. CPS are also used to automate, control, monitor, and enhance manufacturing

TABLE 3.1  
*CPS Applications and their Cyber and Physical Worlds*

CPS Applications	Physical Parts	Cyber Parts	Benefits
Medical CPS	Patients, illnesses, and drugs	Monitoring and controlling patient health status and data	Timely and accurate patient monitoring and treatment
Smart Buildings	Buildings, temperature, lighting, air quality and building residents	Monitoring environmental conditions and energy usage and implementing algorithms to control equipment	Reducing energy consumption and maintaining required quality of living
Smart Grid	Electricity, fuel, power generators, power distribution networks, and consumer devices	Real-time monitoring and controlling energy productions and consumptions	Optimizing energy utilization, reducing overload risks and energy waste
Gas & Oil Pipelines Monitoring & Control	Oil, gas, pipeline networks, pipeline physical status including pressure, and temperature	Monitoring the pipeline status and controlling the pipeline operations	Maintaining health and operations of the pipeline and reducing the impact of failures and accidents
Smart Water Networks	Water, water distribution networks and their status, water storage, water pumps, and water generators	Monitoring and controlling the process of transferring and storing water and its quality and its usage	Reducing water loss, optimizing water production and utilization, and enhancing water quality
Vehicular Safety Applications	Vehicles, vehicles mechanical and electronic devices, roads, drivers and passengers	Real-time algorithms to monitor vehicles status and control vehicles to avoid accidents and optimize flow	Reducing the possibility of accidents, congestion and traffic violations
Intelligent Traffic Light Controls	Traffic lights, vehicles and their positions, roads, pedestrians, and intersections	Real-time algorithms to monitor traffic status and to control traffic lights	Reducing traffic delays, minimizing vehicles travel times, increasing vehicles average velocity, and enhancing the prioritization for emergency vehicles movements
Self-Driving Vehicles	Vehicles equipment, Vehicles energy, passengers, resources, position, roads, roads traffic lights and signs, and neighboring vehicles	Smart algorithms to automate vehicle driving and to maintain driving safety	Reduce transportation costs, optimize traffic flow and enhance safety
Manufacturing Control and Monitoring	Manufacturing requirement, raw materials, products, workers, and warehouses	Real-time algorithms to monitor and control production processes, equipment and material utilization, and product quality	Optimize production and maintenance and enhance product quality
UAV	UAV equipment and energy, position, and operational space	Algorithms to control the UAV, optimize movement and collaboration	Enhanced operations and safely and achieving operational goals
Energy Efficiency in Data Centers	Data center equipment, energy supplies including renewable energy, equipment and buildings temperatures, ventilation and air conditioning	Algorithms to monitor the status of the data center and control servers operations and temperatures in the center	Reducing energy consumption and maintaining the good health of the equipment
Wind Farms	Wind, wind turbines, control equipment, and energy distribution and storage equipment	Algorithms to monitor and control wind turbines and the produced power and to optimize energy production and storage	Maximizing power generation and enabling integration with other systems such as smart grids
Hydropower Plants	Hydraulic engines, transducers, power meters, electric power generators, water level, water flow, energy storage and distribution equipment	Algorithms to monitor and control the power generation, distribution and storage, and the water flow	Maximizing power generation and enabling integration with other systems such as smart grids
Greenhouse Efficient Control	plants, climatic conditions, soil, ventilation, carbon dioxide, water, and heating, cooling, and ventilation equipment	Algorithms to regulate greenhouse climate, optimize resources utilization and maximize production	Enhancing plants growths and produce production and quality, and optimizing resources (e.g. water and energy) utilization

processes [50]. Table 3.1 summarizes the CPS applications in terms of their physical and cyber parts and their benefits.

**4. CPS Applications Challenges.** We realize based on the discussion of the various applications of CPS in Section 3 that there are some common challenges facing developing these applications. The main challenges include:

1. Real-Time operations: Most CPS applications need to function in real-time to deliver usable information. This includes real-time sensing, communication, processing, decision making, and actions. In most applications, the earlier we receive status information and generate the required controls, the better

- results we can achieve. For example, a self-driving vehicle, requires immediate knowledge of the route changes, traffic conditions and traffic light status. Any delays may lead to catastrophic results such as not being able to respond correctly to a changing traffic light or another vehicle not behaving correctly.
2. **Heterogeneous Devices:** CPS applications are built with multiple heterogeneous devices like sensors, actuators, microcontrollers, and communication and storage devices. In addition, they operate in heterogeneous physical systems and environments. This leads to complications in implementing the various controls and integrating the different components as each will need its own models and software components. As a result, introducing new equipment or changing current ones, which occur frequently, will have to involve changes in the software being used.
  3. **Limited Capability Devices:** Some CPS need to use devices with limited capabilities and remote functions. This is mainly due to the current limitations on available devices or to reduce the cost of the CPS. These devices usually have limited wireless communication, processing, and storage capabilities and many also may have limited power sources. Careful design is needed to include these devices as components CPS. This will require complex algorithms to manage, operate and control these devices within their limitations.
  4. **Distributed Processing:** CPS applications require distributed processing and decision making to enhance their operations. In addition, some applications need to use parallel processing for faster operations. This introduces three major challenges: one is the different types and models of communication to be used given the heterogeneity of the devices and connections being used; another is the delays and reliability of the communication, which must be addressed efficiently to enable stable and reliable operations; in addition to the security and privacy of the system and information being used as they travel over various communication channels in the system.
  5. **Security and Privacy:** As most CPS support distributed critical applications, there are high security and privacy concerns. These are introduced mainly by the distributed nature of the system and its components. In addition, many CPS are used within the context of critical and private domains, where data must be protected for various reasons. Therefore, the security and privacy of the information and software must be protected.
  6. **Reliability and Fault Tolerance:** Many CPS applications are critical applications; therefore, they need to be reliable and highly available. It is important to implement mechanisms to increase reliability, which vary depending on the types of devices and software being used and the operational parameters of these devices. In addition, data integrity and correctness must be preserved to achieve high reliability of the system. These systems should be able to operate effectively even when faults occur and should be able to detect and resolve different types of faults without negatively impacting the physical systems.
  7. **Communication:** Special communication requirements and capabilities are needed by some CPS applications among their devices and subsystems. These requirements may include real-time support, highly available and reliable communication, high-bandwidth and efficient information exchange. Some large-scale and highly distributed CPS applications may also need optimized communication mechanisms for collecting and distributing data among the systems components. In addition, different components of a CPS may require different modes of communication and others could be designed to adapt the communication modes to available resources.
  8. **Mobility:** Some CPS applications involve mobile devices that need to be efficiently and securely connected with the rest of the system. This may require actively managing the mobile devices as they change location. Thus, the software used should be able to discover, monitor and control mobile devices as they travel within, enter or exit the geographic boundaries of the system.
  9. **CPS Devices Locations:** CPS applications that involve mobile devices, need to be aware of the mobile devices locations to optimize their operations and correctly achieve the applications objectives. For many CPS applications, knowledge of the instantaneous location of the mobile devices help optimize operations using these devices. For example, when a UAV is available in a specific area, it could be given a specific task relevant to that area to accomplish before moving out of range.
  10. **Power Limitation:** Some CPS components are used in remote locations or in areas where no constant power sources are available. These devices can operate for a limited time and may not be easily

replaceable. Thus, the CPS design should focus on extending the life of these components using efficient and power-aware software, hardware, and communication protocols. The software needs to optimize their operations and limit access to their resources using other alternatives when possible.

11. **Integration with Other Systems:** Some CPS applications require integration with other computing systems or other CPS. For example, integrating CPS applications in neighboring vehicles to enable collaborative safety applications across multiple vehicles or integrating some components with powerful resources and platforms such as cloud-based services offering processing power, storage, and data services. This extends some of the challenges discussed such as security and privacy, reliability and real-time support in addition to the introduced complexity of ensuring seamless interoperations between the systems.
12. **Intelligent Decisions:** Most CPS applications need to make intelligent decision to optimize their operations. This may involve including intelligent algorithms such as data mining algorithms in the CPS design. Many of these require large amount of data and intensive computations, as a result, efficient mechanisms to enable these operations need to be included in the application and could also lead to the need for integration with more powerful systems such as the cloud to support these operations.
13. **Context Awareness:** Some CPS cannot properly or efficiently function without knowing the context of their systems resources, physical environment and general domain. This will require the knowledge of specific information such as power levels, communication and processing status, and other external physical contexts utilized for the operations of CPS. Therefore, the software used must be able to collect context data, organize it and make it available for the CPS application to be used effectively.
14. **Big Data:** Some CPS have a large number of sensors that continuously generate data resulting in huge data sets [80]. At the same time, these systems cannot be optimized well to achieve the objectives of the applications without analyzing this collected data. This may require having a good system infrastructure capable of dealing with the challenges of storing and processing big data as well as algorithms capable of the required analysis.

Table 4.1 summarizes the challenges for different CPS applications. These challenges make developing CPS applications without using proper tools and middleware very difficult. Developers need to be aware of the specific challenges for the application being developed. They need to find effective and efficient methods and algorithms to address these challenges and implement them within the software. As a result, the application must include complex components addressing the actual domain of the application in addition to the challenges imposed by the CPS environment, architecture and components.

As we can see not all challenges apply to all CPS, and for this reason, we identify the challenges that apply to CPS with respect to their application domains. Table 4.1 shows the different application domains of CPS and what challenges are of importance to that domain. As an example, we use smart buildings as an emerging type of CPS. Smart buildings feature Building Management Systems (BMS) that provide monitoring and control to the various elements and subsystems, in fulfillment of preset objectives. Typical objectives of smart buildings are minimizing energy consumption and maximizing occupants comfort. The challenges relevant for smart buildings are the following: Real-time operations, as buildings host occupants, whose well-being depends on the non-stop correct operations of the system; Heterogeneous Devices, as smart buildings incorporate a variety of devices to support their daily functions [64] ; Security and Privacy Support, as the security and privacy of occupants data regarding their interactions with the buildings systems need to be protected against any threats or attacks and must be kept private [23] ; Reliability and Fault Tolerance, as some buildings are safety-critical (e.g. hospitals or power stations) where the probabilities of failures should be minimized [47] ; Integration with Other Systems, which is especially important when buildings are considered elements of smart cities and smart grids, where they will need to function flawlessly in connection to other systems (e.g. traffic or smart grid, but also other smart buildings BMS); Intelligent Decisions, as BMS goals are to optimize buildings performance; and Context Awareness, as smart buildings are not isolated and their environment plays a vital role in their performance [33].

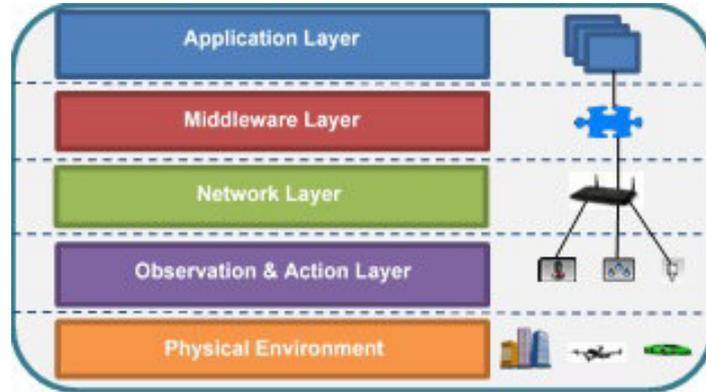
**5. Middleware for CPS.** Middleware platforms can provide important support for developing and operating CPS applications. The availability of a suitable middleware platform that provides solutions for the challenges discussed earlier will not only enhance the CPS development process, but also enhance reusability

TABLE 4.1  
*Different Challenges of Different CPS Applications in the General Cases*

	Real-Time Operations	Heterogeneous Devices	Limited Capabilities	Distributed Processing Need	Security and Privacy Requirements	Reliability & Fault Tolerance	Special Communication	Mobility Challenge	Location-Based Requirement	Power Limitation Challenge	Integration with Other Systems	Intelligent Decision Requirement	Context Awareness Challenge	Big Data
Medical CPS	x	x	x		x	x	x	x		x	x	x	x	x
Smart Buildings	x	x			x	x					x	x	x	x
Smart Grid	x	x		x	x	x	x				x	x	x	x
Gas & Oil Pipelines Monitoring & Control	x	x	x	x	x	x	x			x	x		x	
Smart Water Networks	x	x	x	x	x	x	x			x	x	x	x	x
Vehicular Safety Applications	x	x			x	x		x	x			x	x	
Collaborative Vehicular Safety App.	x	x		x	x	x	x	x	x		x	x	x	
Self-Driving Car	x	x		x	x	x		x	x		x	x	x	
Manufacturing	x	x			x	x					x		x	x
UAV	x	x	x		x	x		x	x	x		x	x	
Collaborative UAVs	x	x	x	x	x	x	x	x	x	x	x	x	x	
Energy Efficiency in Data Centers	x	x		x	x	x			x	x	x	x	x	
Wind and Hydro Power Plants	x	x		x	x	x					x		x	
Greenhouse Efficient Control		x	x		x	x				x		x	x	

and maintainability, reduce risks, and reduce the overall cost of developing and maintaining CPS applications. With a suitable middleware platform designed to support CPS applications, the developer can focus more on developing the main functions of the CPS applications rather than expending huge time and effort in implementing solutions and codes to solve the general and common issues such as reliability and security. The specialized middleware for CPS can be designed to include a set of services that provide solutions for various common CPS challenges that exist for any type of CPS applications. These services can be used by developers to implement different CPS applications. The developers can use middleware APIs for these services to utilize their functions and features and integrate them with the required specific functions of the CPS. In addition, the middleware can enable reuse of any previously implemented module in new CPS applications. This will also reduce the time and efforts needed to test new modules as the reusable modules have already been tested and approved for use. The middleware will also enable maintainability by allowing easy changes for any modules in the system. Any software module or service can be changed or replaced by better implemented one if it uses the same interface. This will also enable easy change to some hardware components in CPS applications to better hardware components.

The architecture of CPS implemented with middleware has five layers as shown in Figure 5.1. These layers are physical environment layer, observation and action layer, network layer, middleware layer, and application layer. The physical layer is the lowest layer in the CPS architecture and it consists of all the physical environment or parts of the CPS that will be monitored or controlled by CPS applications. This layer can include parts like vehicles, streets, buildings, the human body, machines, energy, etc. The second layer from the bottom is the observation and action layer. This layer is responsible for observing the current status of the physical environment and acting on the environment by changing its current state. This layer will have different types of sensors such as temperature sensors, movement sensors, cameras, sound sensors, pressure sensors, GPS, RFID, bar code readers as well other types of sensors and monitors to observe the current status of the monitored environments. In addition, it also has a number of actuators such as hydraulic, pneumatic, electric, thermal,

FIG. 5.1. *CPS Layers*

and mechanical actuators. These actuators are responsible for moving or controlling mechanisms, systems, or environments and changing their current physical states. The third layer is the network layer which is responsible for collecting the data observed by the observation and action layer and transporting it to the upper layer, the middleware layer. It is also responsible for transporting information from the middleware layer to the observation and action layer. The network layer can deal with simple peer-to-peer networks or with multi-hop networks of any scale. The network can be wired, wireless, or both. Generally, this layer is responsible for enabling information exchange between the middleware layer and the observation and action layer.

The Middleware layer, which is the focus of this paper, is responsible for providing a set of services to support implementing and operating CPS applications. These services can be basic communication services that enable the exchange of messages and information among CPS components or value-added services such as real-time support, action validation support, reliability and fault tolerance support, mobility support, location-based support, and security attacks detection services. In addition, these services can be very advanced smart services that are controlled by specifying high-level policies and global objectives for the whole systems operations. Examples of these services can be context awareness services, multiple CPS collaboration services, intelligent decision services, self-adaptive services, self-resilient services, and self-protected services. These middleware services can be selectively used by the top layer, the application layer to develop and operate CPS applications specific to the needs of the CPS in use. The Middleware layer can abstract the heterogeneity and distribution of the lower layers including the physical environments and hide their technical details. This feature can reduce the complexities of CPS applications. The CPS applications developers can have a set of advanced application interfaces (APIs) provided by the middleware to use the provided services. These services provide ready-made solutions for common challenges facing different CPS applications. The developer will not need to spend a significant time developing, implementing, and testing new code for these common challenges. As a result, using the right middleware for CPS can reduce the risks of having bugs in the CPS applications as the available middleware services will be developed more carefully and used by multiple applications. The middleware approach will allow for pre-designed services modeled specifically for the devices and environments in use, thus allowing for better utilization of these components. In addition, adding, changing or removing components will not require major changes in the application software as the middleware makes it possible to perform these tasks while maintaining a consistent API for the applications. From the developer perspective, a device performing a specific type of activity is merely a service available with known APIs to access it. The actual details of its operations and technical specifications are hidden and of no concern to the application developer. At a higher level, since we view the features needed as services, it becomes possible to implement new services, update current services, and provide alternative implementation of some services allowing the application developer to pick and choose the suitable services for the application being developed. Furthermore, this allows for updates, increments and adjustments to currently used services without having to rebuild the applications using them.

We classify middleware systems that can support CPS applications into three types. These three types are based on their abstraction levels and their support functions. Each of these types can support different

programming models to develop and operate CPS applications. Here discuss the main characteristics and functions of the three types and the relationships among them.

**5.1. Communication Middleware for CPS Applications.** This type of middleware mainly enables and facilitates communication among heterogeneous and limited resources CPS components. This includes offering services for efficient unicast communication and efficient group communication for broadcasting, multicasting, and data collection. It can provide basic security mechanisms to be used by developers to protect the communications among CPS components. It also provides basic mechanisms to communicate with other systems through message passing and/or remote procedure calls. While this type of middleware enables communication among CPS components, the developers need to solve and write codes for many other challenges mainly due to its limited scope and functions. The developers also need to handle the details of distributed processing in CPS. One possible approach for a communication middleware is to use a customized massing passing model which can meet the CPS communications challenges.

**5.2. Value-Added Services Middleware for CPS.** In this type, a number of services can be provided and used part of the CPS applications adding features and value to these applications. These services can be real-time support, monitoring, validation, reliability, fault tolerance, mobility support, location-based support, security attacks detection services, and service-level integration with other systems. A higher level of programming models for distributed and parallel processing can be used with this middleware to allow developers to implement CPS applications using the available services. This middleware type includes a resource manager and scheduler to enable implementing the services and also to allow developers to define available resources and detailed policies guiding the use of the available services. One of the most suitable middleware architectures to use here is the service-oriented middleware [19].

**5.3. Advanced Services Middleware for CPS.** Here we describe advanced services and smart support components, as an addition to the services described as Value-Added Services Middleware. These include autonomous resource discovery and management, context awareness support, multiple CPS collaboration support, and intelligent decision support. In addition, smart services such self-adaptive, self-resilient, and self-protected services can be offered. The developers in this case will resort to advanced high-level abstracted programming models to write the CPS applications instead of conventional programming languages. The developers can also specify high-level policies and global objectives for the whole system to operate on. Examples of these programming models in related systems such as WSNs are Kairos [34] and Cougar [81].

These three middleware types differ in the level of abstraction and how much time and effort needed by the developers to implement new CPS applications. The developers need more time and effort to develop new applications with the communication middleware alone as they need to deal with individual components in the CPS, while they need less time and effort with the value-added services middleware as they will use the available services to implement their applications. Furthermore, they need much less time with the advanced services middleware as they will deal with high-level policies and global objectives and the middleware will map these into implemented services. However, this type of middleware is very difficult to implement as it needs to self-handle most of the CPS common challenges.

**6. Middleware Challenges for CPS.** Although middleware platforms provide many advantages for implementing and operating CPS applications, developing such middleware platforms is also challenging. In this section, we discuss some of the challenges of designing and implementing such middleware platforms:

1. **Enabling Smooth and Efficient Integration:** CPS middleware should enable smooth and efficient integration among all CPS heterogeneous components. Components in any CPS can be developed and implemented by several manufactures. While some CPS components are implemented such that they support standard interfaces for interaction with other components, others are implemented without supporting any interface standards. One of the roles of the middleware is to enable the integration among these heterogeneous components and ensuring proper integration with the various available interfaces. In addition, these components may have different communication capabilities and operational standards. They may use different communication protocols, different communicate rates, different synchronization capabilities, and different security capabilities. The role of the middleware here is to enable the efficient and smooth integration among all used communication models in use and enabling seamless communication between the devices and components used by the CPS applications.

Furthermore, the CPS middleware should enable the integration with other systems outside the boundaries of the CPS environment such as cloud and fog computing and other CPS systems.

2. **Supporting Advanced Communication Schemes:** Some CPS applications cannot be efficiently offered without using advanced communication schemes such as the Publish/Subscribe communication scheme [29] which is needed for reducing communication overhead in large scale systems with a large number of sources of information or events such as large numbers of sensors and control components or actuators using the produced information and events. Another example of these advanced communication schemes is the store and forward communication scheme [45] which is needed for large mobile applications with discontinuous communication links among their mobile CPS components. These applications can be collaborative UAVs or collaborative vehicular applications. These advanced communication schemes are usually not supported by the current traditional communication technologies while they can be effectively offered by the middleware layer and used by the CPS applications.

3. **Resource Management:** The need for providing real-time, reliable, fault tolerant, power efficient, and automatic management by the CPS middleware require having an efficient and smart resource manager that can provide essential features that provision such services. These features include efficient resource discovery, monitoring, and control for both limited and unlimited capabilities components in the CPS. In addition, the resource manager should be supported by an efficient scheduler for utilizing these resources as well as QoS support for both processing and communication. The middleware should be able to map the performance requirements of different applications into system level parameters that can configure the underlying system to achieve these specified performance requirements. However, designing the scheduler and mapping processes for CPS middleware is challenging given the large number of resources and the variety of specifications, functions and requirements of each of these resources. Moreover, the design of fast and optimal or near-optimal algorithms for resource allocation and adaptation can also be very challenging. This is due to heterogeneity of the resources, the limited capabilities of some of the CPS components, and the high CPS application requirements such as safety, security, scalability, and sustainability. Furthermore, the scheduler needs to make tradeoff decisions among communication, computations, monitoring, and control to achieve the CPS applications objectives. These tradeoffs can be challenging and requires smart decisions for optimal or semi-optimal resource scheduling.

4. **Secure Middleware Services:** As most CPS are considered critical applications and the middleware is the backbone for integrating the CPS components and enabling the CPS operations, then all middleware services should be secure. Any security leakage in the offered services can be utilized to gain unauthorized access to the CPS applications, which imposes many risks such as suspension the operations of the CPS applications or altering the operations of the CPS applications to unsafe operations thus resulting in damages in the corresponding physical environment or systems or interfering with normal operations leading to serious problems.

5. **Global Reference Time Support:** Operations in many CPS applications cannot be correctly done without having a global reference time to be used by all components of CPS to order and synchronize events and actions in the CPS [77]. This is one of the requirements in some CPS applications for ensuring safe and accrued operations. While a physical reference time support is offered in some new hardware and networks, it is not provided and supported by many CPS components. In this case, the CPS middleware can provide a logical global reference time support to be used for CPS applications. This requires designing accurate middleware-based global reference time support that meets the CPS challenges.

6. **Load Balancing:** Different load balancing aspects are needed in constrained resources CPS to enhance the utilization of CPS resources, to meet applications requirements, to enhance performance, and to increase the sustainability of the system. The load balancing aspects can be related to balancing distributed and parallel processing to enhance response time or to meet with time constraints, balancing the power consumption to enhance the sustainability of the system, and balancing the communication traffic to enhance throughput and data transfer times, in addition to balancing overall CPS operations.

7. **Scalability support:** some CPS applications involve a large number of components and extend over large physical environments or systems [73]. Examples of these applications are smart grids and gas and oil pipeline monitoring and control systems, where a large number of sensors and actuators are used covering extensive geographic areas. Designing such large-scale systems requires good middleware support to deal with large number of widely distributed components as well as high communication traffic generated from these CPS

components. In addition, some CPS applications could expand over time and include more components, services and sub systems. This growth could affect the overall performance of the system if not designed to scale well.

8. Supporting autonomous operations for complex CPS applications: Many CPS applications are considered complex systems incorporate many components that interact with each other for monitoring and controlling physical environments and systems. These complex systems can be in a huge number of different states at any point of time. It is generally extremely difficult to develop code to handle all these states effectively and in a timely manner. Having middleware that supports autonomous operations such as self-adaptive, self-resilient, and self-protected services [28] can relax implementing and operating these complex CPS applications. However, providing such services can be very challenging as it is not easy to predict all possible states and situations early in the design process. In addition, many of these self-x properties require complex algorithms and in some cases some intelligent components to be handled correctly. Moreover, testing verifying and validating these services when implemented is also challenging due to the large and complex set of possible combinations of events and states that could trigger them.

**7. Current Research Efforts.** There is some ongoing research to customize existing middleware platforms or design new middleware services to fit with the CPS challenges. One of these important challenges is supporting real-time operations in CPS. Real-time support requires provisions from the operating systems, resource managers and networks. The requirements and an architecture for a CPS middleware supporting these provisions was proposed in [31]. In addition, different real-time challenges for diverse scenarios were proposed. An example of these proposals is the approach to solve the real-time issue for aperiodic events in distributed CPS using a reconfigurable real-time middleware [82]. Another proposal is RDDS which is a publish/subscribe- middleware architecture developed to enable timely and reliable sensor data dissemination in highly unpredictable CPS environments [42]. There are also some research efforts dedicated to address the heterogeneity challenges in CPS. A middleware that provides interoperability between heterogeneous mobile devices in CPS was proposed in [76]. Furthermore, developing portable middleware services for heterogeneous CPS was proposed in [55]. Other research efforts were conducted to investigate reliability, security, safety, and fault tolerance in CPS. As an example, the main role of middleware in facilitating robust and resilient CPS was studied in [27] while a reliable, safe, and secure run-time platform for CPS was proposed in [51]. Moreover, a time-triggered middleware architecture that offers fault tolerance and dynamic reconfiguration at run-time taking into consideration the available system resources of the underlying infrastructure was proposed in [61].

Another group of research efforts were dedicated to investigating issues in large-scale CPS. The design, development, testing, and operations of a large-scale CPS are more complex compared to other CPS. This is due to higher heterogeneity, unreliability, unpredictability, complexity, and security requirements of large-scale CPS [46]. Therefore, large-scale CPS are very complicated to develop and operate without relying on support from advanced middleware services. Advanced middleware services can provide interoperability, reliability, QoS, and security mechanisms to satisfy the needs of large-scale CPS. In this regard, an efficient middleware for supporting distributed query processing in large-scale CPS was proposed in [26]. The work in [69] investigated developing a middleware on WSN for large-scale CPS. The aim of this middleware is to automatically achieve optimal sensor node configuration, bandwidth provisions, fault handling, and re-configuration in reaction to new missions and new added devices. In addition, a virtualized network platform for testbed of large-scale CPS was proposed in [16]. The requirements of virtual platform and networks for very large-scale CPS that expanded globally are investigated in [15].

A service-oriented approach to build middleware platforms for CPS was instigated into a number of research projects. This approach can solve many CPS challenges. A service-oriented middleware architecture to expose CPS devices to the Web was addressed in [37]. In addition, a service-oriented middleware for fog and cloud integrated CPS was proposed in [59] while a service-oriented approach to address fault tolerance in CPS was proposed in [21]. Another solution is a real-time service-oriented architecture middleware to monitor the performance and reserve resources in advance for CPS services in process to ensure its real-time achievability [52]. In the service-oriented middleware approach, system resources are viewed as a set of services to be used to develop CPS applications. One of the main advantages of this approach is the flexibility feature of extending the middleware itself to include new and more advanced services to support CPS applications as they develop. In addition, it provides the flexibility to add more devices, components, and services as the CPS grows or more

features are needed.

Some middleware platforms were also developed for specific applications or to solve specific issues in these applications. Examples include a middleware support for continuous monitoring of water distribution systems [40], a service-oriented middleware for smart grids [83, 54], a service-oriented middleware for collaborative UAVs, an event-driven middleware for smart buildings [63], an adaptive middleware for context-aware smart home applications [39], a middleware architectural framework for vehicular safety [72], an interoperable middleware platform for medical CPS [65]. Generally, these efforts provide solutions to specific issues in CPS rather than addressing the generic model that can support various features and apply to different CPS applications. More work is needed to address the general issues facing most, if not all, CPS applications and offer middleware platforms that can be adapted and used for several applications.

**8. Open Issues.** Based on the studied CPS applications and proposed middleware solutions, there are still a number of CPS middleware issues that need more research and deeper investigation leading to usable and effective solutions. The following is a discussion of some of these open issues:

1. **Generic Middleware Architectures:** There are several proposals for middleware architectures that are suitable of some CPS while they are unsuitable for others. These offer specialized solutions applicable to the application domain they target only. Developing a generic middleware architecture for all CPS applications is needed as they share several challenges that can be addressed effectively and reused for all CPS applications. The availability of such middleware architecture can provide a base for enhancing many solutions for many of the challenges in CPS. This also provides a common platform where features and services can be added, updated, enhanced or redesigned to benefit all applications. It can also enhance the development processes for CPS applications by offering more flexible design, implementation, testing, and reusability features. Moreover, using the same middleware platform for multiple CPS applications, will allow these applications to integrate easily and interoperate to achieve larger objectives.

2. **Resource Management:** Various efforts have been conducted to develop resource management techniques in traditional distributed systems. However due to the unique challenges of CPS, it is not easy to adapt these traditional techniques for CPS. Any adaptation of traditional resource management techniques into the context of CPS needs careful consideration of the efficiency, flexibility, and scalability of these techniques. Issues such as heterogeneity, varying capabilities, mobility, time constraints, and connectivity introduce more challenges for the resource manager. More investigation and proposals are needed for middleware-based resource management techniques in CPS.

3. **Middleware Security Support:** It is impossible to build secure and reliable CPS applications without considering the security and privacy aspects of these systems. Middleware can provide a number of security services for CPS. However, very limited research and development efforts were conducted in this regard. Mean-time, a number of security middleware solutions were developed for other types of distributed systems such as ubiquitous applications [17]. These solutions provide a useful base for developing and operating ubiquitous applications. Similarly, security middleware solutions are needed for CPS.

4. **Middleware Safety Support:** As faults in CPS can cause severe and in many cases irreparable physical damages, middleware platforms should provide runtime environments for CPS applications that offer support for the safety of the CPS and its physical environment. More investigation is needed to find effective ways to utilize middleware for safety support. This can include, for example, developing middleware-based runtime validations and including fault detection and correction mechanisms within the middleware functions.

5. **Middleware Sustainability Support:** CPS are usually designed for critical applications that should live for a long period of time. However, some CPS devices have limited power, while others may have limited operational life. Over time many devices will need to be replaced for various reasons and the replacements may or may not be of the same type or capabilities. Designing any CPS solutions without considering the energy limitations, wear out probabilities and replacement devices will reduce the operational life span of the CPS applications. Careful designs are needed for all aspect of CPS to extend their life. This includes efficient energy aware features to conserve energy, unified interfaces to support device swaps and updates, and seamless integration of new devices and components. While this design can be extremely difficult, middleware platforms for CPS should assist in this regard. However, there are no comprehensive investigations toward developing middleware based solutions for sustainability support for CPS.

**9. Conclusions.** CPS applications are becoming an integral part of many environments and cover a diverse set of application domains. As a result, their design, development and operations have become complex and time consuming. The different types of applications have unique requirements and impose different challenges for the application developer. One method to help leverage some of the challenges and support the development process is to use middleware platforms. As discussed in this paper, middleware can provide various essential features and services for the CPS applications. In addition, it can also provide value-added features that enhance the operations and capabilities of the CPS applications. Unfortunately, such middleware, if designed to cover all needed aspects and functionalities of all types of CPS applications, will itself become too complex and difficult to design. We discussed the various challenges of middleware for CPS including the support for advanced communication schemes, effective and efficient resource management and load balancing, scalability, global time reference, security and autonomy. As discussed, many have addressed some of these challenges either on a generic basis for a single feature or specifically addressing one or a group of similar CPS applications. Unfortunately, there is a lot more to be done before a comprehensive middleware platform can be designed to support a large variety of CPS applications. A number of open issues need to be addressed and incorporated in a middleware approach. One of the main issues is the design of a generic middleware platform that can support different types of CPS applications. Other issues include the security and safety of the middleware in addition to resource management. Finally there is the issue of sustainability of the CPS application and its components. These issues have not been adequately addressed and require more efforts to create usable and efficient solutions. In our future work we intend to further investigate these issues and create possible approaches to address them. For example, we are looking into models for resource management for CPS environments that include stationary, mobile and limited resource devices.

## REFERENCES

- [15] S. AHN, C. YOO, S. LEE, H. LEE, AND S. J. KIM, *Implementing virtual platform for global-scale cyber physical system networks*, International Journal of Communication Systems, 28 (2015), pp. 1899–1920.
- [16] S. W. AHN AND C. YOO, *Wip abstract: Virtual network platform for large scale cps testbed*, in Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems, IEEE Computer Society, 2012, p. 214.
- [17] J. AL-JAROODI, I. JAWHAR, A. AL-DHAHERI, F. AL-ABDOULI, AND N. MOHAMED, *Security middleware approaches and issues for ubiquitous applications*, in Computers and Mathematics with Applications, Special Issue on Advances in Cryptography, Security and Applications for Future Computer Science, Vol. 60, No. 2, 2010, pp. 187–197.
- [18] J. AL-JAROODI AND N. MOHAMED, *Middleware is still everywhere!!!*, Concurrency and Computation: Practice and Experience, 24 (2012), pp. 1919–1926.
- [19] ———, *Service-oriented middleware: a survey*, Journal of Network and Computer Applications, 35 (2012), pp. 211–220.
- [20] J. AL-JAROODI, N. MOHAMED, I. JAWHAR, AND S. LAZAROVA-MOLNAR, *Software engineering issues for cyber-physical systems*, in IEEE International Conference on Smart Computing (SMARTCOMP), 2016.
- [21] P. ALHO AND J. MATTILA, *Service-oriented approach to fault tolerance in cps*, Journal of Systems and Software, 105 (2015), pp. 1–17.
- [22] S. ALI, S. B. QAISAR, H. SAEED, M. F. KHAN, M. NAEEM, AND A. ANPALAGAN, *Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring*, Sensors, 15 (2015), pp. 7172–7205.
- [23] P. ARJUNAN, N. BATRA, H. CHOI, A. SINGH, P. SINGH, AND M. B. SRIVASTAVA, *Sensoract: a privacy and security aware federated middleware for building management*, in Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, pp. (2012), pp. 80–87.
- [24] C. BERGER AND B. RUMPE, *Autonomous driving - 5 years after the urban challenge: The anticipatory vehicle as a cyber-physical system*, In Proceedings of the INFORMATIK, 2012 (2012), pp. 789–798.
- [25] M. CHAQFEH AND N. MOHAMED, *Challenges in middleware solutions for the internet of things*, in proc, 2012 Int'l Conference on Collaboration Technologies and Systems (CTS), 2012.
- [26] A. CUZZOCREA, J. CECILIO, AND P. FURTADO, *An effective and efficient middleware for supporting distributed query processing in large-scale cyber-physical systems*, in International Conference on Internet and Distributed Computing Systems, Springer International Publishing, ed., 2014, pp. 124–135.
- [27] G. DENKER, N. DUTT, S. MEHROTRA, M. O. STEHR, C. TALCOTT, AND N. VENKATASUBRAMANIAN, *Resilient dependable cyber-physical systems: a middleware perspective*, Journal of Internet Services and Applications, 3 (2012), pp. 41–49.
- [28] X. DONG, S. HARIRI, L. XUE, H. CHEN, M. ZHANG, S. PAVULURI, AND S. RAO, *Autonomia: an autonomic computing environment*, In the, 2003 (2003), pp. 61–68.
- [29] P. T. EUGSTER, P. A. FELBER, R. GUERRAOU, AND A. M. KERMARREC, *The many faces of publish/subscribe*, ACM computing surveys (CSUR), 35 (2003), pp. 114–131.
- [30] Y. P. FALLAH, C. HUANG, R. SENGUPTA, AND H. KRISHNAN, *Design of cooperative vehicle safety systems based on tight coupling of communication, computing and physical vehicle dynamics*, in Proc. of the 1st ACM/IEEE International Conference on Cyber-Physical Systems, ACM, 2010, pp. 159–167.

- [31] M. GARCA-VALLS AND R. BALDONI, *Adaptive middleware design for cps: Considerations on the os, resource managers, and the network run-time*, In Proceedings of the, 14 (2015).
- [32] L. GONDA AND C. E. CUGNASCA, *A proposal of greenhouse control using wireless sensor networks. in computers in agriculture and natural resources*, American Society of Agricultural and Biological Engineers, 229 (2006).
- [33] T. GU, H. K. PUNG, AND D. Q. ZHANG, *A service-oriented middleware for building contextaware services*, Journal of Network and Computer Applications, 28 (2005), pp. 1–18.
- [34] R. GUMMADI, O. GNAWALI, AND R. GOVINDAN, *Macro-programming wireless sensor networks using kairos*, in International Conference on Distributed Computing in Sensor Systems, Springer Berlin Heidelberg, ed., 2005, pp. 126–140.
- [35] V. GUNES, S. PETER, T. GIVARGIS, AND F. VAHID, *A survey on concepts, applications, and challenges in cyber-physical systems*, TIIS, 8 (2014), pp. 4242–4268.
- [36] L. GURGEN, O. GUNALP, Y. BENAZZOUZ, AND M. GALLISSOT, *Self-aware cyber-physical systems and applications in smart buildings and cities*, in Proc. of the Conference on Design, Automation and Test in Europe, 2013, pp. 1149–1154.
- [37] D. D. HOANG, H. Y. PAIK, AND C. K. KIM, *Service-oriented middleware architectures for cyber-physical systems*, International Journal of Computer Science and Network Security, 12 (2012), pp. 79–87.
- [38] X. HU, T. H. CHU, V. C. LEUNG, E. C. H. NGAI, P. KRUCHTEN, AND H. C. CHAN, *A survey on mobile social networks: Applications, platforms, system architectures, and future research directions*, IEEE Communications Surveys & Tutorials, 17 (2015), pp. 1557–1581.
- [39] M. C. HUEBSCHER AND J. A. MCCANN, *Adaptive middleware for context-aware applications in smart-homes*, In Proceedings of the, 2 (2004), pp. 111–116.
- [40] M. IQBAL AND H. B. LIM, *A cyber-physical middleware framework for continuous monitoring of water distribution systems*, in Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, ACM, 2008, pp. 401–402.
- [41] M. S. KAMAL, S. PARVIN, K. SALEEM, H. AL-HAMADI, AND A. GAWANMEH, *Efficient low cost supervisory system for internet of things enabled smart home*, in Conference on Communications Workshops (ICC Workshops), Ieee International, ed., IEEE, 2017, pp. 864–869.
- [42] W. KANG, K. KAPITANOVA, AND S. H. SON, *Rdds: A real-time data distribution service for cyber-physical systems*, IEEE Transactions on Industrial Informatics, 8 (2012), pp. 393–405.
- [43] S. KARNOUSKOS, *Cyber-physical systems in the smartgrid*, in 9th IEEE International Conference on Industrial Informatics (INDIN), 2011, pp. 20–23.
- [44] S. KARTAKIS, E. ABRAHAM, AND J. A. MCCANN, *Waterbox: A testbed for monitoring and controlling smart water networks*, in Proc. of the 1st ACM International Workshop on Cyber-Physical Systems for Smart Water Networks, 2015.
- [45] A. KESTING, M. TREIBER, AND D. HELBING, *Connectivity statistics of store-and-forward intervehicle communication*, IEEE Transactions on Intelligent Transportation Systems, 11 (2010), pp. 172–181.
- [46] W. T. KIM, I. G. CHUN, S. H. LEE, H. Y. LEE, AND J. M. KIM, *Wip abstract: From design to operation of a large-scale cps*, in Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems, Ieee Computer Society, ed., 2012.
- [47] S. LAZAROVA-MOLNAR, H. R. SHAKER, AND N. MOHAMED, *Reliability of cyber physical systems with focus on building management systems*, in proc. 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC), 2016.
- [48] E. A. LEE, *Cyber physical systems: Design challenges*, In Object Oriented Real-Time Distributed Computing (ISORC), pp. (2008), pp. 363–369.
- [49] I. LEE, O. SOKOLSKY, S. CHEN, J. HATCLIFF, E. JEE, B. KIM, A. KING, M. MULLEN-FORTINO, S. PARK, AND A. ROEDERER, *and k.k.*, in Challenges and research directions in medical cyber-physical systems, In Proc. of the IEEE, 100(1), 2012, Venkatasubramanian, pp. 75–90.
- [50] J. LEE, B. BAGHERI, AND H. A. KAO, *A cyber-physical systems architecture for industry 4.0-based manufacturing systems*, Manufacturing Letters, 3 (2015), pp. 18–23.
- [51] S. S. LIM, E. J. IM, N. DUTT, K. W. LEE, I. SHIN, C. G. LEE, AND I. LEE, *A reliable, safe, and secure run-time platform for cyber physical systems*, in 2013 IEEE 6th International Conference on Service-Oriented Computing and Applications (SOCA), IEEE, 2013, pp. 268–274.
- [52] K. J. LIN AND M. PANAH, *A real-time service-oriented framework to support sustainable cyber-physical systems*, in 2010 8th IEEE International Conference on Industrial Informatics, IEEE, 2010, pp. 15–21.
- [53] S. P. LIN AND N. F. MAXEMCHUK, *The fail-safe operation of collaborative driving systems*, Journal of Intelligent Transportation Systems, 20 (2016), pp. 88–101.
- [54] J. F. MARTNEZ, J. RODRIGUEZ-MOLINA, P. CASTILLEJO, AND R. DE DIEGO, *R., “middleware architectures for the smart grid: survey and challenges in the foreseeable future.”* Energies, 6 (2013), pp. 3593–3621.
- [55] K. MECHITOV AND G. AGHA, *Building portable middleware services for heterogeneous cyber-physical systems*, In Proceedings of the Third International Workshop on Software Engineering for Sensor Network Applications, pp. (2012), pp. 31–36.
- [56] L. MICLEA AND T. SANISLAV, *About dependability in cyber-physical systems*, In, 2011 (2011), pp. 17–21.
- [57] N. MOHAMED, J. AL-JAROUDI, I. JAWHAR, AND S. LAZAROVA-MOLNAR, *A service-oriented middleware for building collaborative uavs*, Journal of Intelligent & Robotic Systems, 74 (2014), pp. 309–321.
- [58] N. MOHAMED, J. AL-JAROUDI, S. LAZAROVA-MOLNAR, AND I. JAWHAR, *Middleware to Support Cyber-Physical Systems*, in proc. IEEE Int’l Performance Computing and Communications Conference (IPCCC 2016), Las Vegas, Nevada, USA, 2016.
- [59] N. MOHAMED, S. LAZAROVA-MOLNAR, I. JAWHAR, AND J. AL-JAROUDI, *Towards service-oriented middleware for fog and cloud integrated cyber physical systems*, in IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), IEEE, 2017, pp. 67–74.

- [60] M. M. MOLLA AND S. I. AHAMED, *A survey of middleware for sensor network and challenges*, in ICPP 2006 Workshops, IEEE, 2006.
- [61] A. NOGUERO, I. CALVO, AND L. ALMEIDA, *A time-triggered middleware architecture for ubiquitous cyber physical system applications*, in International Conference on Ubiquitous Computing and Ambient Intelligence, Heidelberg, Berlin, 2012, Springer, pp. 73–80.
- [62] L. PAROLINI, B. SINOPOLI, B. H. KROGH, AND Z. WANG, *A cyber-physical systems approach to data center modeling and control for energy efficiency*, in Proceedings of the IEEE, 100(1), 2012, pp. 254–268.
- [63] E. PATTI, A. ACQUAVIVA, M. JAHN, F. PRAMUDIANTO, R. TOMASI, D. RABOURDIN, J. VIRGONE, AND E. MACII, *Event-driven user-centric middleware for energy-efficient buildings and public spaces*, IEEE Systems Journal, 10 (2016), pp. 1137–1146.
- [64] T. PERUMAL, A. R. RAMLI, C. Y. LEONG, K. SAMSUDIN, AND S. MANSOR, *Middleware for heterogeneous subsystems interoperability in intelligent buildings*, Automation in Construction, 19 (2010), pp. 160–168.
- [65] J. PLOURDE, D. ARNEY, AND J. M. GOLDMAN, *Openice: An open, interoperable platform for medical cyber-physical systems*, in Conference on Cyber-Physical Systems (ICCPS), Acm/ieee International, ed., IEEE, 2014, pp. 221–221.
- [66] J. M. PORTOCARRERO, F. C. DELICATO, P. F. PIRES, B. COSTA, W. LI, W. SI, AND A. Y. ZOMAYA, *Ramses: a new reference architecture for self-adaptive middleware in wireless sensor networks*, Ad Hoc Networks, 55 (2017), pp. 3–27.
- [67] R. R. RAJKUMAR, I. LEE, L. SHA, AND J. STANKOVIC, *Cyber-physical systems: the next computing revolution*, in Proc. of the 47th Design Automation Conference, ACM, 2010, pp. 731–736.
- [68] M. A. RAZZAQUE, M. MILOJEVIC-JEVRIC, A. PALADE, AND S. CLARKE, *Middleware for internet of things: a survey*, IEEE Internet of Things Journal, 3 (2016), pp. 70–95.
- [69] N. REIJERS, Y. C. WANG, C. S. SHIH, J. Y. HSU, AND K. J. LIN, *Building intelligent middleware for large scale cps systems*, in 2011 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), IEEE, 2011, pp. 1–4.
- [70] K. RMER, O. KASTEN, AND F. MATTERN, *Middleware challenges for wireless sensor networks*, ACM SIGMOBILE Mobile Computing and Communications Review, 6 (2002), pp. 59–61.
- [71] A. A. SALKHAM, R. CUNNINGHAM, A. GARG, AND V. CAHILL, *A collaborative reinforcement learning approach to urban traffic control optimization*, in Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, 2008, Volume 02, pp. 560–566.
- [72] K. SARAVANAN, A. THANGAVELU, AND K. RAMESHBABU, *A middleware architectural framework for vehicular safety over vanet (invanet)*, in First International Conference on Networks and Communications (NETCOM'09), IEEE, 2009, pp. 277–282.
- [73] D. C. SCHMIDT, J. WHITE, AND C. D. GILL, *Elastic infrastructure to support computing clouds for large-scale cyber-physical systems*, in 2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), IEEE, 2014, pp. 56–63.
- [74] J. SHI, J. WAN, H. YAN, AND H. SUO, *A survey of cyber-physical systems*, in 2011 International Conference on Wireless Communications and Signal Processing (WCSP), IEEE, 2011.
- [75] S. SRIDHAR, A. HAHN, AND M. GOVINDARASU, *Cyber-physical system security for the electric power grid*, in Proceedings of the IEEE, 100(1), 2012, pp. 210–224.
- [76] J. SUN AND Y. ZHANG, *A middleware for highly dynamic distribution in cps environment*, in Proceedings of the 3rd International Conference on Context-Aware Systems and Applications, Social-Informatics and Telecommunications Engineering, 2014, ICST (Institute for Computer Sciences), pp. 169–172.
- [77] Y. TAN, S. GODDARD, AND L. C. PEREZ, *A prototype architecture for cyber-physical systems*, ACM Sigbed Review, 5 (2008), p. 1.
- [78] M. TANG, X. DAI, J. LIU, AND J. CHEN, J., *“towards a trust evaluation middleware for cloud service selection,”* Future Generation Computer Systems, 74 (2017), pp. 302–312.
- [79] Q. TANG, S. K. S. GUPTA, AND G. VARSAMOPOULOS, *Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach*, IEEE Transactions on Parallel and Distributed Systems, 19 (2008), pp. 1458–1472.
- [80] S. M. TONNI, M. Z. RAHMAN, S. PARVIN, AND A. GAWANMEH, *Securing big data efficiently through microaggregation technique*, in IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), IEEE, 2017, pp. 125–130.
- [81] Y. YAO AND J. GEHRKE, *The cougar approach to in-network query processing in sensor networks*, ACM Sigmod record, 31 (2002), pp. 9–18.
- [82] Y. ZHANG, C. GILL, AND C. LU, *Reconfigurable real-time middleware for distributed cyber-physical systems with aperiodic events*, in The 28th International Conference on Distributed Computing Systems (ICDCS'08), IEEE, 2008, pp. 581–588.
- [83] L. ZHOU AND J. J. RODRIGUES, *Service-oriented middleware for smart grid: Principle, infrastructure, and application*, IEEE Communications Magazine, 51 (2013), pp. 84–89.

*Edited by:* Amjad Gawanmeh

*Received:* May 30, 2017

*Accepted:* Nov 6, 2017