IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 23, 2020, accepted October 23, 2020, date of publication October 28, 2020, date of current version November 13, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3034390*

# Environmental-Perception Modeling and Reference Architecture for Cyber Physical Systems

**NIANYU LI[1], KUN LIU[1], ZHENGYIN CHEN[1], AND WENPIN JIAO[2]**
[1]Key Laboratory of High Confidence Software Technologies, Peking University, Beijing 100871, China
[2]Department of Computer Science and Technology, School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China

Corresponding author: Wenpin Jiao (jwp@sei.pku.edu.cn)

This work was supported in part by the National Basic Research Program of China (973) under Grant 2015CB352200 and Grant 2016YFB000105, and in part by the National Natural Science Foundation of China under Grant 61620106007.

**ABSTRACT** In cyber-physical systems, physical and software components are deeply intertwined, blurring the boundaries between the cyber and physical worlds. Perceiving environmental information is a prerequisite for a cyber-physical system to be reliable and adaptive to the environment. However, the intrinsically open and dynamic nature of the environment brings challenges to the systematic realization of environmental-perception. This article regards environmental-perception as a first-class entity with self-maintenance and abstracts it into three levels: data capture, context awareness, and situation identification. Then, a five-tier reference architecture is proposed, which not only explicitly defines the perception ability at different abstraction levels, but also provides for the storage and management of environmental information, thus facilitating information reuse and the adaptation ability of cyber-physical systems. We evaluate the reference architecture with a case study on a widely used smart room environment, demonstrating its broad applicability and the ability to reuse environmental information. We also performed a survey on the proposed reference architecture to reveal its industrial motivations and benefits.

**INDEX TERMS** Environmental perception, cyber physical systems, self-adaptation, reference architecture, context awareness, situation identification, information reuse.

## I. INTRODUCTION

In cyber-physical systems (CPS), ‘‘physical and computation components are deeply intertwined, each operating on different spatial and temporal scales, exhibiting multiple and distinct behavioral modalities, and interacting with each other in many ways that change with context.’’[1] CPSs blur the boundary and realize the integration between the cyber and physical worlds [1].

With the rapid growth in heterogeneity and complexity of CPS, many uncertainties cannot be entirely considered or predicted by developers during the design process, bringing significant challenges to the design and implementation of CPS [2]. The intrinsically open and dynamic nature of the environment requires that a CPS must be robust to unexpected conditions and adaptable to changes in the environment [3].

[1]US National Science Foundation, Cyber-Physical Systems (CPS), https://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm

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

The first step to adaptation is to monitor the environment and perceive the dynamics and environmental information [4], [5], that CPS can use to dynamically modify its behavior, structure, and parameters to continuously satisfy users' goals, including performance, security, and fault management. Therefore, *environmental-perception*, as a prerequisite for a CPS to be reliable and adaptive, has been assigned importance in recent years. Despite preliminary progress, there are still several challenges on *environmental-perception* not completely solved.

First of all, a CPS has a limited capacity for perception and computing in many cases [6]. It is impossible to deploy an infinite number of devices to perceive all the environmental-information by the system itself [7]. Also, the information required by the system may need processing based on the directly perceived information, such as filtering, aggregation, organization, and prediction. For example, the location information of a user may be aggregated by multiple sensors in several corners of a room. This may overwhelm the

CPS with limited computational capability. Second, a CPS would not only run in a specific environment. However, the module of perceiving the environment is often dealt with implicitly or in an ad hoc manner [8], [9], which tightly binds this module of a CPS to the environment. When the CPS meets a newly deployed environment, the perception module or component needs to be redesigned and implemented, which is a poor engineering practice and inevitably leads to high costs. Thirdly, as the environment is increasingly dynamic and active with humans involved, the way of acquiring environmental information could be passive, active or even predictive [10], [11]. Equipping the CPS with a variety of perception means will inevitably increase the complexity of the *environmental-perception* implementation.

An environment provides the surrounding conditions for a system to exist. Essentially, the environment is sometimes affected by the system's behavior or can be changed independently by itself. *Environmental-perception*, as an intermediary between the environment and its deployed CPSs, should be isolated from the CPS, like an operating system, providing the CPS an appropriate interface by shielding low-level perception details and making it easier for the system to access and use the information in various deployed environments [12]. Moreover, there may be multiple cyber-physical systems realizing different functionalities in an environment. Ideally, the information in an environment is preferably managed for efficient sharing and reuse among different systems. Also, the life cycle of *environmental-perception* covers many aspects, including acquisition, preprocessing, storage and update, modeling and representation, use and reasoning of environment information. In the absence of a reference architecture, building a widely applicable *environmental-perception* framework - so complex and heterogeneous information can be easily expressed, reasoned, and reused - is still a challenge to be explored.

To deal with the aforementioned challenges, we first define *environmental-perception* as a first-class entity. The term "first-class entity" stresses the fact that *environmental-perception* is a building block that encapsulates its own clear-cut responsibilities [8] of environmental information management and maintenance independent of CPSs. Then we divide *environmental-perception* into three levels: data capture, context awareness, and situation identification. As a first-class entity, *environmental-perception* can independently collect the low-level information, and combine prior knowledge of aggregation or fusion to obtain more abstracted information, which can be reused by multiple CPSs deployed in the environment and shields the CPSs from tedious detection and low-level information processing details. By defining the context configuration, one no longer needs to consider the perception details when designing and developing the system. Instead, the system acquires the context pushed from the *environmental-perception* entity. The system only needs to focus on extracting semantic interpretations from the context in its deployment environment to identify the state of interests for the system (i.e., situation) at the application level, thus realizing the system's perception of various environments. Then, we propose a reference architecture from the idea of ISO network protocols to implement the above three levels. The architecture provides not only perception approach for environment information at different levels, but also a means of storage and management for the perceived information and a unified interface for the system to access the information that the system needs and uses to infer whether it can continuously meet its goal.

The rest of the paper is structured as follows: Section 2 summarizes the existing approaches of perceiving environmental-information and presents our new approach regarding the *environmental-perception* as a first class entity existing independently; Section 3 introduces three levels of *environmental-perception* and defines its constituents respectively; Section 4 specifies a reference architecture for implementing the three levels with reference to ISO protocols; Section 5 presents a smart room environment example to illustrate how CPS perceives the environment information through our proposed framework and reference architecture; Section 6 includes a survey to further reveal the industrial motivations and benefits of the reference architecture; Section 7 details some related work; Section 8 makes some concluding remarks on this article and points out our future work.

## II. ENVIRONMENTAL-PERCEPTION: AN OVERVIEW

The environment is the external world for its deployed systems and exists objectively and independently; perceiving environmental information is a prerequisite for a CPS to be reliable and adaptive. There are still several challenges that need to be addressed for a CPS to perceive environmental information, including 1) how to deal with complex environment information; 2) how to filter out useful information from all the information in the environment; and 3) how to further understand useful information to facilitate making effective adaptation decisions. In this section, after introducing some common existing approaches for perceiving the environment, we propose separating the responsibilities of *environmental-perception* from the CPS as an independent first-class entity to address the aforementioned challenges.

### A. EXISTING APPROACHES FOR PERCEIVING THE ENVIRONMENT

The first common approach for a system to perceive environmental information is to deploy the sensors or devices in advance, directly gathering low-level information and processing it in a customized way. On this basis, environmental information is used to reason about whether the system realizes its goal in the application environment and whether system adaptation is required. This approach is very primitive, and perceiving the environment is generally implemented implicitly or in an ad hoc manner in the system. The second approach is to structure the process of perceiving the environment in a standardized and organized manner. One branch is to make perception implementation in a hierarchical structure, dividing the perception into several tiers with each tier corresponding to its specific functionalities. Generally, the lower the tier, the more detailed the information that is directly collected from the real environment; the higher the tier, the more refined and abstracted the information that is processed and recognized by the system. The other branch is to separate perception concerns based on objects or perception purposes (e.g., perceiving services, users, and tasks). The third approach encapsulates the environment-related responsibilities (i.e., perception, data processing, and data management) into an independent building block irrespective of the remaining part of the system. This is also known as the first-class entity, further emphasizing the *environmental-perception* as an essential part of the system.

In these approaches, *environment-perception* is a part of and indivisible from the implementation of the whole system and tightly bound to the environment where the system is deployed. For a changing deployment environment, the way the information is collected and processed may change. For example, to determine whether a user is in front of a desk, the system needs to gather distance information from sensors in the office and aggregate it with the location of the desk. If in a new office with different positions of deployed sensors and the desk, the distance collected and the processing function in this system need to be changed accordingly. This inevitably leads to partial changes in the relevant part perceiving the information with the changing deployment environment. Ideally, the implementation of the system should be reused for a wide range of deployment environments without modification. In addition, some information may be needed and perceived by multiple systems deployed in an environment. Each system using additional devices to collect it will result in a waste of resources. The information in an environment should be reused by multiple systems deployed therein.

### B. ENVIRONMENTAL-PERCEPTION AS A FIRST-CLASS ENTITY

To deal with the shortcomings of existing approaches, we propose regarding *environmental-perception* as a separate first-class entity instead of a part of the system. As a separate first-class entity, *environmental-perception* does not belong to any system but coexists with the environment. Furthermore, *environmental-perception* is divided into three levels: data capture, context awareness, and situation identification to address three aforementioned challenges respectively as shown in Figure 1. Responsible for information management and maintenance for an environment and separated from the system itself, *environmental-perception* facilitates the reusability of the environmental information among multiple deployed CPSs and the applicability of a CPS to be deployed in various application environments without modification.

On the data capture level, *environmental-perception* can independently capture the underlying information through all devices in the environment, and process it with prior knowledge to form various data items, which can be reused by multiple systems. Context awareness, as the second layer,

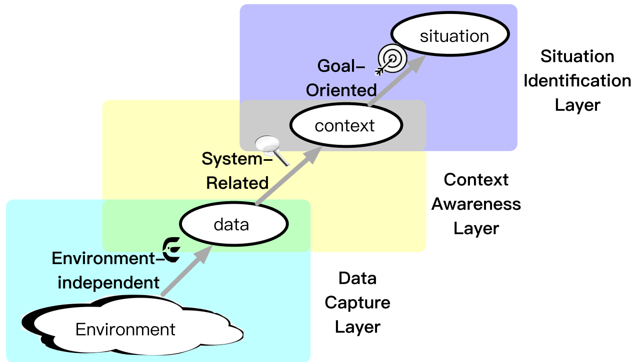



**FIGURE 1.** Three Levels of ***Environmental-Perception***.

screens out the information according to the needs of the systems through their context configurations. Situation identification, as the third layer, identifies the information of interests of the system and assigns the meaning from the perspective of the application and system goals. By adopting the hierarchical structure, a system itself no longer needs to consider the details of the perception or processing of the low-level details. In this way, *environmental-perception*, as a separate first-class entity, will manage the hugely complex real-world information for CPSs, thus facilitating their applicability to different deployment environments. The formalization of these three levels will be introduced in Section III while the reference architecture for the implementation of *environmental-perception* as a separate first-class entity is presented in Section IV.

## III. THREE LEVELS OF ENVIRONMENTAL-PERCEPTION

In this section, we will formally define the three levels of *environmental-perception*: data capture, context awareness and situation identification.

### A. DATA CAPTURE

Figure 2 is the conceptual model composed of concepts and their associations on the data capture layer. The environment of cyber-physical systems covers different aspects (i.e., human society, cyberspace, and the physical world) and inhabits a set of attributes. To collect and make use of environment information, different sensors in CPSs will measure the attributes in the environment and transform them into the required forms for information transmission, processing, storage, and display. For example, current smart cities embed a constellation of sensors, wireless devices, and actuators on the physical infrastructures connected through sensor networks to form the foundations of a city's smart infrastructures.

Raw values *RV* refer to information representing environmental attributes, including binary and continuous attributes, and collected directly by sensors deployed in the environment. Binary forms are the simplest type with true or false values, analogous to the switch on a desk lamp. Temperature and absolute humidity are two typical examples of continuous form. In addition, errors may occur during raw value acquisition and transmission; the *validation* function

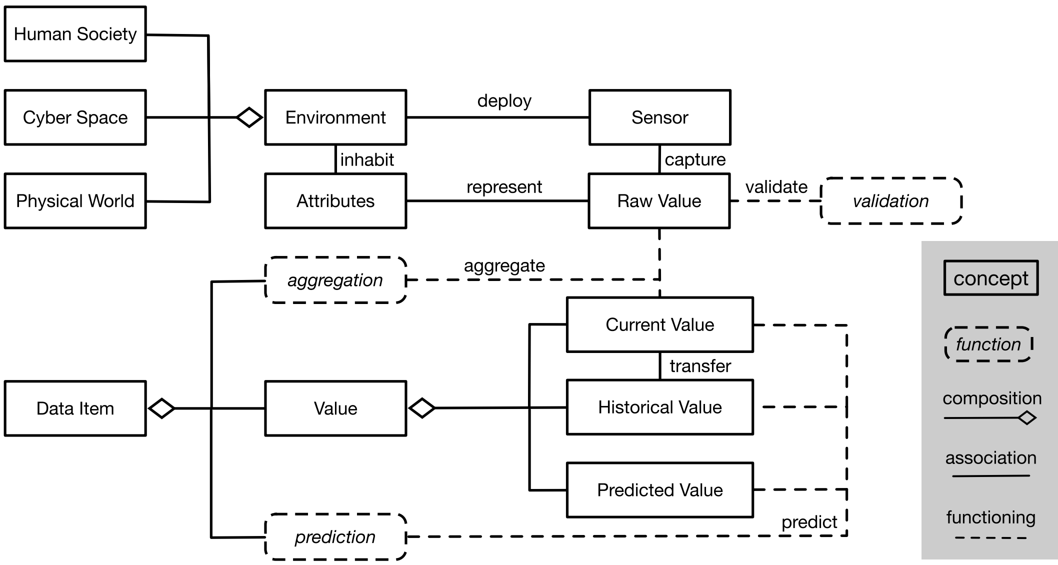



**FIGURE 2.** **Conceptual Model of Environment-Independent Data Capture.**

$vld : RV \rightarrow Bool$ mapping raw values to boolean values validates and discerns which deviate from the normal range at a specific time. Only when the output is true can the current attribute value be valid. Otherwise, the sensor has to be examined or the raw value needs to be recollected.

Raw values might be too low-detailed and trivial to properly represent environmental information a system requires, while the data item might have relatively coarse granularity. An *environmental-perception* entity provides various data items, and each data item is defined as a triple with its value and two functions, as follows.

*Definition 1:* A data item is defined as a triple $DI =< Val, agg, pre >$

- The value $Val = \{d_0, d_1, \ldots, d_t, d_{t+1}, \ldots\}$, where $t$ is the current time stamp, is composed of historical values $d_0, d_1, \ldots, d_{t-1}$, current value $d_t$, and predicted values $d_{t+1}, \ldots$.
- The aggregation function $agg : 2^{RV} \rightarrow Val.d_t$ abstracts and synthesizes one or more raw values from sensors as inputs and generates the current value for a data item. The current value of a data item is often combined with prior knowledge and obtained by aggregating different raw values. For instance, data of relative humidity needs to be aggregated by the raw values from the temperature sensor and from the humidity sensor.
- The prediction function $pre : (Val.(d_0, \ldots, d_t)) \rightarrow Val.(d_{d+1}, \ldots))$ inputs the historical values and the current value, and outputs the prediction results either in one period $d_{t+1}$ or multi-period $d_{t+1}, \ldots, d_{t+h}$ in the next $h$ horizon. Data prediction is an important step to further understand environmental information, such as predicting the whereabouts of the user in advance.

Raw values are the results of direct observation of the environment, while the data items are abstracted, synthesized, and predicted on the basis of those raw values. For the first layer of the independent first-class entity – *environmental-perception* – data capture collects attribute values through all sensors in the environment, validates and aggregates the acquisition value to generate coarse-grain data items with prior knowledge, and makes predictions. This layer preprocesses environmental information independent of the system to provide data support and facilitate the use of and access to environmental information for the second layer.

### B. CONTEXT AWARENESS

Figure 3 depicts the conceptual model of the context awareness and the situation identification layers. A CPS usually consists of the system goals, the perceived environment state and the system state to estimate whether the goals are violated, and adaptation rules to achieve goals when violations occur with the input of current system state and perceived environment state. A perceived environment state is derived from the understanding of the needed surrounding information, such as the number of available servers and users, which is the required context.

The term "context" is a concept that spans across many different domains without a universally-acknowledged definition. Therefore, researchers are allowed to define context such that it is most suitable for a particular domain [13]. For instance, in natural languages, the context is the parts of a discourse that surround a word or passage and can throw light on its meaning; in operating systems, the context is the set of data used by a task (which may be a process or thread) that must be saved to allow a task to be interrupted [12]; in pervasive computing, a widely accepted definition is any information that can be used to characterize the situation of an entity [14]; in service computing, the context is the set of all input and output parameters that may affect the service [15]. In the framework of *environmental-perception*, we define the context of a cyber-physical system as *a set of data in the environment acquired or used by the system for decision-making and achieving goals.*

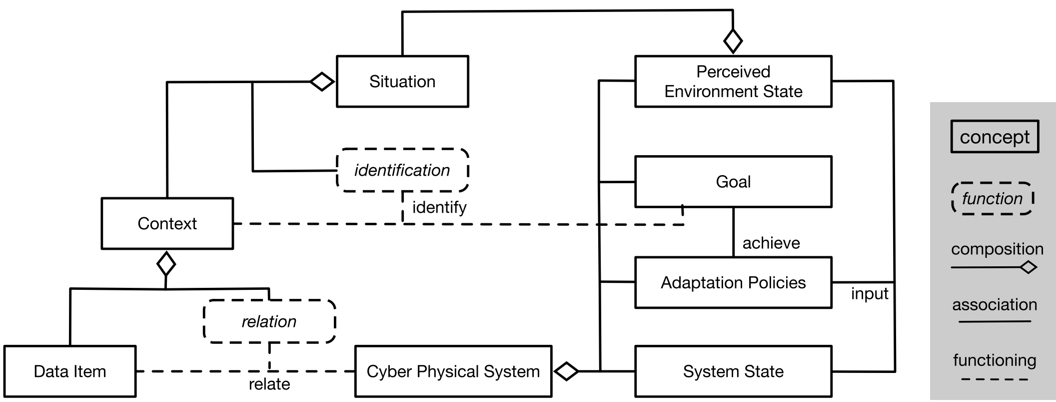



**FIGURE 3.** Conceptual Model of System-Dependent Context Awareness and Situation Identification.

*Definition 2:* The context for a system is a triple: $Cnt =< Sys, \ rel, \ CDI \ >$

- `Sys` is the system label since the context is system-dependent.
- The relation function $rel \ : \ (DI, Sys) \ \rightarrow \ bool$ denotes whether a certain data item is required by the system. For example, in an urban environment, there might be an intelligent transportation system, an intelligent medical system, an intelligent government system, etc. A data describing an vehicle location information may be related to the intelligent transportation system $rel(di_{vehicle} \quad sys_{transportation\ system}) = True$, but not to the intelligent government system $rel(di_{vehicle}, sys_{government\ system}) = False$.
- $CDI$ is the set of data items in need: $CDI = \{x|x \in \{DI_1, DI_2, \ldots, DI_n\} \cap rel(x, Sys) = true\}$.

For a system, the data from the data capture layer is intuitively divided into two subsets: one containing information that has nothing to do with the system, the other one describing the external information required by the system (i.e., context) which could be declared via the context configurations in the system.

In addition, the temporal aspect of adaptation is divided into two dimensions: reactive and proactive [16]. Reactive adaptation only starts to function after being triggered by an event, such as a change in resources or a drop in performance. On the contrary, proactive adaptation is defined as modifications of an application performed before an application becomes executable. Context-awareness can be enhanced by the proper use of context prediction – short-term and long-term prediction of relevant data, such as, predicting the user actions or the mobility of nodes, and steps can be taken in advance to make proactive adaptation possible.

Overall, the second layer of *environmental-perception* selectively screens out the data items from the data capture layer for each system according to its need.

### C. SITUATION IDENTIFICATION

Situation identification is built on top of context awareness and aims to infer situations out of context on a high-abstracted level. The situation is a subjective concept and identified as the external semantic interpretations of sensor data in pervasive computing [17]. Semantic interpretations mean that the situation assigns meaning to data, while externality means situation identification is from the perspective of system application and goals of the system, instead of data perspective. Taking an e-commerce website as the example, if the server of the website has an average response time of 5 seconds to the users' requests, it is interpreted that the current users' response delay is very high for this website application, though 5 seconds is not chronologically long. However, for such an application, 2 seconds may have lead to deteriorated user experience.

As shown in the above example, the interpretation depends on the domain knowledge and interests expressing the goals. In the framework of *environmental-perception*, we define the situation of a system as *an interesting condition with external semantic interpretations for a part of data items in the system context from the perspective of system goals*. The situation identification layer generalizes the context and understands the meaning behind it.

*Definition 3:* A situation of a system is defined as a triple $Sit =< Sys, \ idt, \ cndt \ >$.

- `Sys` is the system label since the situation is system-dependent.
- The identification function $idt : (2^{Cnt.CDI}, Sys.goal) \rightarrow cndt$ receives a subset of data items from the context and extracts them to a higher level condition depending on the system application and goal with domain knowledge.
- An interesting condition $cndt$ is identified as $cndt = idt(sCnt, Sys.goal)$ where $sCnt \subset Cnt$. The condition for a system is the interpretation of entity status (e.g., *computer being used*) or human behavior (e.g., *sitting* or *standing*).

Even with the same data items in the context, different systems will interpret them in different situations. For example, based on the context data of locations for several users, the situation of the user-centered system is in the *meeting* while that of the location-centered system is presented as being *occupied*.

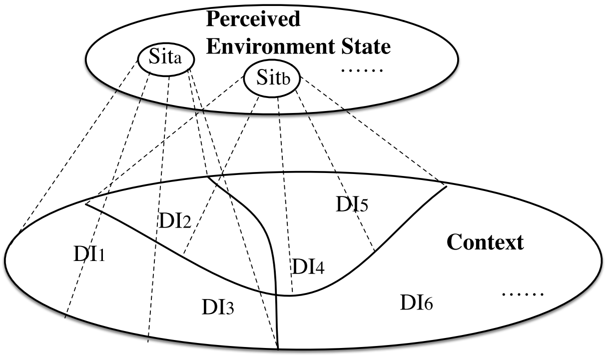



**FIGURE 4.** Situation Identification from Context.

Moreover, as shown in Figure 4, the context of a system could be interpreted as multiple situations. Each situation might be an understanding of partial data items in the context, where data items could be repeatedly combined and interpreted as different situations. For example, a smart office system may summarize the data of several users' locations as *in the meeting*. Meanwhile, the location of a user as well as the on or off status of devices is considered as *projector being used*. For a CPS, it is the situations that make up the perceived environment state.

A situation summarizes a part of context data and obtains the most important information based on the system goal. Situation identification based on context awareness is with a higher-abstracted level; adaptation behaviors can be planned by determining adaption rules with the input of both a system state and a perceived environment state consisting of the situations in which a system is interested.

## IV. REFERENCE ARCHITECTURE FOR ENVIRONMENTAL-PERCEPTION

In *environmental-perception*, the granularity of the raw values that can be detected directly in the environment is relatively low-detailed, while the context that a system needs may be a collection of middle-level information from aggregated and refined low-level information. Moreover, to achieve system goals, high-level information may need to be summarized and semantically interpreted. However, currently, there is a lack of a reference architecture, which should be responsible for information collection and pretreatment serving as a medium for various CPSs, to guide *Environmental-Perception*.

Among the three levels of *Environmental-Perception*, there are not only the functions of *validation*, *aggregation*, and *prediction* that need to be implemented in the data capture layer, but also data classification in the context awareness layer and semantic extraction in the situation identification layer. A good reference architecture distinguishes different functions for easy maintenance and implementation; it should also be independent of the specific environment and system, facilitating reusability of environmental information. As an open standard for information exchange between all kinds of hardware and software, OSI protocols [18], designed by the International Organization of Standardization, do not rely on any specific computer or operating system, nor do they depend on the particular transmission hardware. Through hierarchical segmentation, these protocols promote standardized communication. Each tier is independent and achieves its functions so as to provide flexibility, such as the physical tier responsible for the hardware connections of network, the data link tier compiling data into a form called frame and handling transmission errors, the network tier collecting data and selecting paths from nodes to nodes, the transport tier establishing end-to-end communication, and the application tier providing the interface for user applications.

By referring to OSI protocols, we pioneered a five-tier reference architecture for *environmental-perception*: the facilitator tier, validation tier, pretreatment tier, transport tier, and application tier. To provide a proof-of-concept implementation of our reference architecture to model *environmental-perception*, we realized a prototypical architecture integrating those five tiers as shown in Figure 5. The reference architecture and partial implementation on the smart room case study elaborated in Section V are available in the online appendix [19]. The facilitator tier, validation tier, and pretreatment tier in the architecture together implement the functions in the data capture layer; the transport tier echoes the context-awareness; the application tier corresponds to situation identification. This architecture provides for the perception and management of the environmental information on different levels and provides a unified interface for different CPSs so that systems could use their context and situations to infer whether they meet their goals. In the following, we will elaborate on each tier.

### A. FACILITATOR TIER

In network protocols, the role of the physical tier is to connect machines through physical means, such as cabling and twisted-pair wiring, and transmit the basic signals – 0 and 1. Similarly, the facilitator tier in *environmental-perception* collects the information that can be detected directly in the environment through sensors. This tier, as shown in Figure 6, corresponds to raw values acquisition from sensors and has three dimensions: perceived objects, perception means, and perception mechanisms.

*Perceived Objects.* With cloud computing, the internet of things, and mobile communication as technical support, the cyber-physical systems integrate the physical world, the physical world, and human society. In such a three-dimensional fusion environment, the perceived objects are divided into physical attributes, information attributes, and user attributes.

*Perception Means.* Physical sensors are the facilities that directly measure attributes in the physical world (such as temperature, light, and humidity); cyber attributes (e.g., residual memory) can be monitored by virtual sensors, such as operating systems, virtual machines, application APIs to detect the computing space, or hard-coded and aspect-oriented programming to monitor software [20]; information can also be user-related (such as user preferences), and user input can

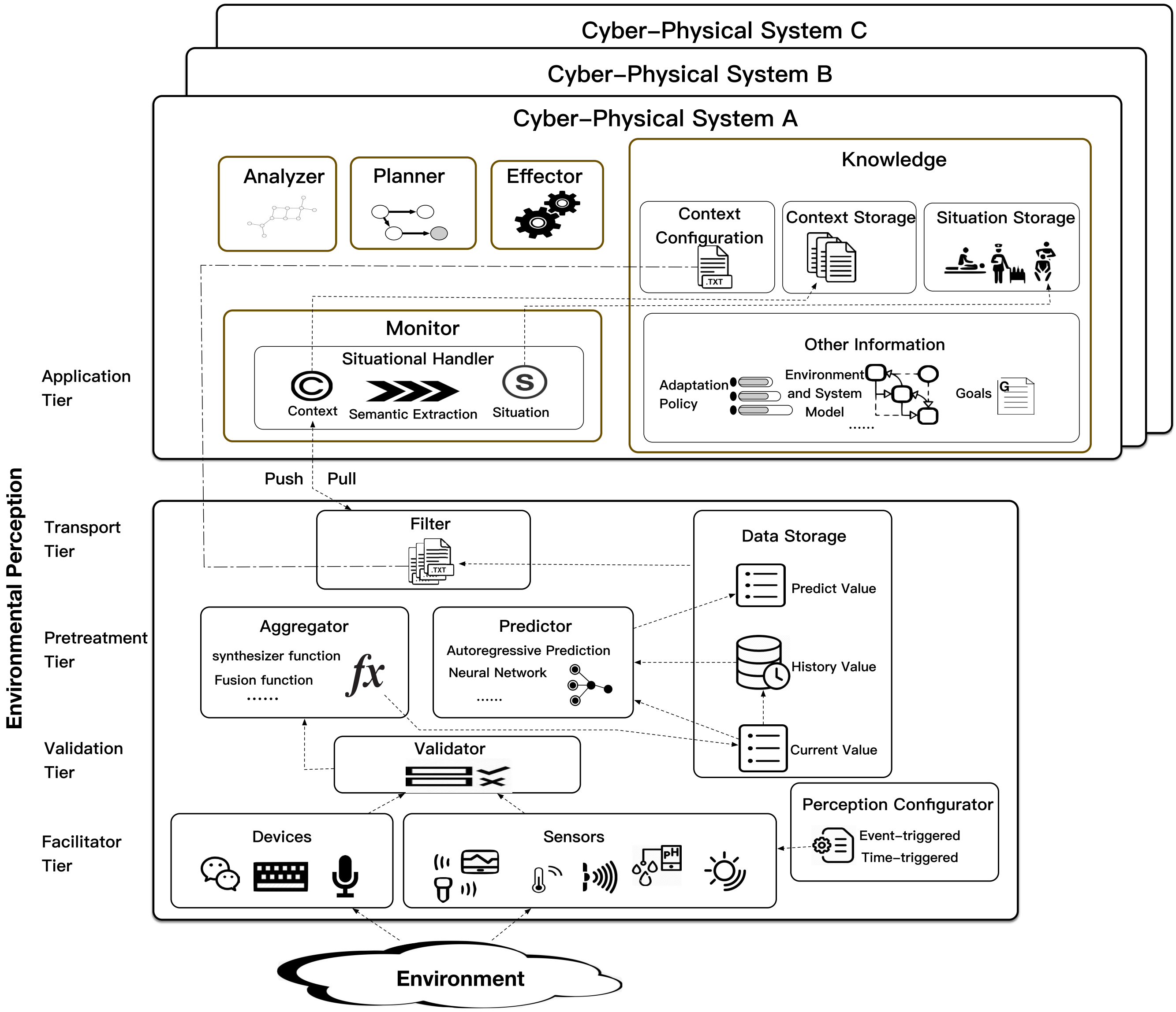



**FIGURE 5.** Reference Architecture for *Environmental-Perception*.

be directly provided by keyboard, touch screen, voice, and other devices. Some common sensors and their data types are summarized in [17].

*Perception Mechanisms.* Raw value acquisition may be active or passive. Devices generally passively or inactively perceive information, that is, only when the user spontaneously provides data can the device be triggered by the events and capture the information. Some sensors have the ability to actively detect and obtain information periodically from the physical world and cyber world. The specific perception mechanism of a sensor is controlled by the *perception configurator* in the environmental perception entity as shown in Figure 5.

### B. VALIDATION TIER

The data link tier in the network mainly establishes a logical connection and applies the error checking function. The environment with three-dimensions is very complex and the number of sensors is inevitably massive. The sensor might break down without being noticed or a mistake could occur in transmission, which introduces dummy values or noise. For example, in a smart room, the false reading of 100 degrees Celsius may harm the accuracy of the aggregated data, and then reduce the adaptability of the system.

In order to avoid the occurrence of the above scenarios, it is necessary to eliminate the false and retain the true value captured by *Validator* shown in Figure 7. A typical example is the consistency checking principle helping to handle the problem of attribute measurement beyond a reasonable range and detect the damaged sensor in time. At present, the physical and the discriminant methods are commonly utilized to check for errors.

### C. PRETREATMENT TIER

The network tier in the OSI protocols is composed of multiple nodes with IP addresses. After data from all sides converges

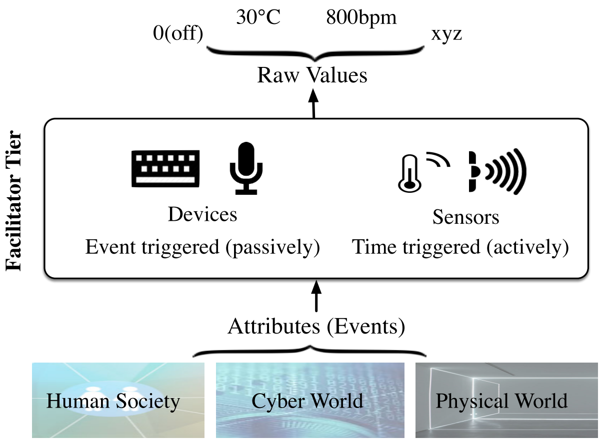



**FIGURE 6.** Facilitator Tier.

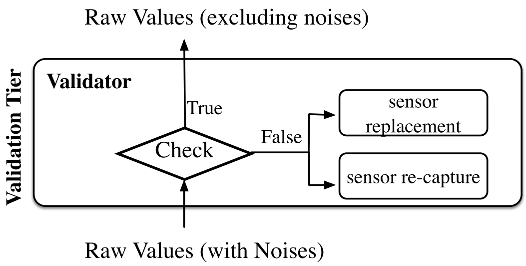



**FIGURE 7.** Validation Tier.

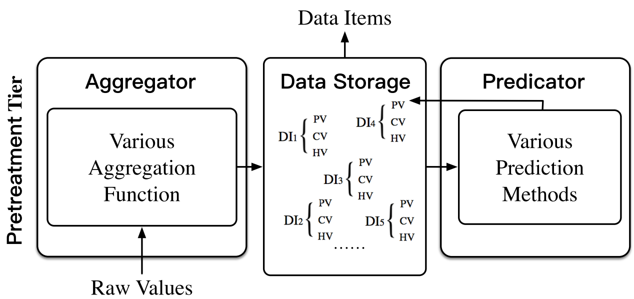



**FIGURE 8.** Pretreatment Tier.

in a distribution center, the layer performs real-time processing and transfers data from nodes to nodes according to their logical addresses. Similarly, the pretreatment tier in the architecture converges all of the low-detailed information in the environment, aggregates it into the data items and makes proper predictions that multiple systems may need.

The pretreatment tier implements the *agg* and *pre* functions in the data capture layer, as shown in Figure 8. The aggregator synthesizes or abstracts raw values into the form the system needs and recognizes (i.e., *DI*), and stores them as the current values of data items in *Data Storage*. The process of obtaining more-abstracted data from low-detailed raw values often involves rules reasoning, machine learning, data mining, and other technologies [20].

To make predictions and make the system history-aware, the historical values of data items need to be stored [21]. Predicator could provide the system with environment information by obtaining the predicted values in the visible future via appropriate methods based on the current value and historical values. Common scenarios [11], [22] requiring prediction and proactive adaptation include 1) reconfiguration, such as installation and updating, loading and unloading libraries, launching new applications, and searching large databases. If the system can predict the needs of these tasks in advance based on environmental information, it can perform tasks ahead of time, avoiding unnecessary delays. 2) power management; unused devices that will not be used recently can be turned off or switched to sleep mode. 3) early warning; data prediction could determine whether the system is about to enter an unexpected state, and take actions accordingly and promptly. 4) planning Aid; such as turning on the air conditioning in advance in a smart home to set the temperature within a certain comfortable range. 5) coordination and cooperation; if the data of multiple users in the user group can be predicted, the interests of the group as a whole are able to be satisfied or balanced.

Different data items need to be predicted within various time frames; some only need the next timestamp; others may need to be predicted for a period of time. Depending on the data items, different prediction methods are adopted. Common methods include the sequence prediction method, Markov chain, neural networks, Bayesian network, branch prediction, trajectory extension method, and expert systems [23], [24].

### D. TRANSPORT TIER

The transport tier transmits data from the source to the destination node in OSI protocols. Like an IP address that could have multiple ports, it is not a rare case where several CPSs are deployed in an environment nowadays. For example, for an office building, there may be a card access control system and a smart parking system; in a city environment, there is not only a smart transportation system but also a smart logistics system or smart education service system.

Essentially, the environment, though influenced by system behaviors, is a separate entity from deployed systems. The reusability of environment information could be greatly enhanced by monolithically capturing, synthesizing, and predicting all kinds of information. For instance, the information about vehicles may not only apply to an intelligent transportation system, but also to the context of an intelligent logistics system. As a layer in the first-class entity, context awareness shields the system from cumbersome perception and processing details; it filters data for the system according to its context configuration as shown in Figure 9 (i.e., the *rel* function determining whether a data item is required by a system). Also, when the deployed environment changes, a system can obtain its context from the new *environmental-perception* entity with the context configuration. Thus, this tier enables information reusability among different systems and enables the system to adapt to different environments.

### E. APPLICATION TIER

Similar to the application tier working at the user end in OSI protocols and interacting with user applications, the

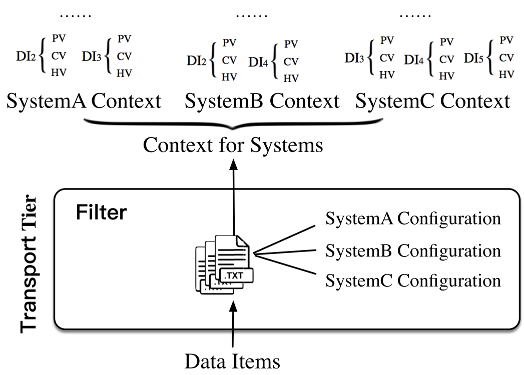



**FIGURE 9. Transport Tier.**

application tier in the *environmental-perception* framework is closest to systems with various functions and goals. On the basis of context awareness, it identifies the most interesting states of each system corresponding to the *idt* function in the situation identification layer, and stores the situation in the knowledge base of the system.

The technologies of situation identification can be divided into two main categories: norm-based methods and learning-based methods. The former usually apply the prior knowledge of experts to construct a situation model and adopt the reasoning engine to infer the appropriate system situation from the current data input. The main technologies of norm-based methods include logical programming, temporal logic, ontology, fuzzy logic, and evidence theory. However, there may be great variability due to factors such as time, location, users, and a complicate environment. This makes the method relying on prior knowledge unusable. Learning-based technologies can be widely used in the complex and dynamic relationship between the context and situation. The mainstream technologies include naive Bayesian, Bayesian networks, hidden Markov models, conditional random domains, decision trees, and neural networks [17].

#### a: ADAPTIVE CPS

*Environmental-perception* is the prerequisite for a CPS to be reliable and adaptive. To achieve self-adaptation, a monitor component of CPS receives environmental information (i.e., the context) from the *Environmental-Perception* entity by system-defined environmental configuration, to find out when, where, and what happens; for example, events of human location movement, connection/disconnection between mobile devices and wireless networks, and configuration/de-configuration of virtual machines, message exchange between devices and virtual machines. Monitor extracts interesting situations from the context, such as *user is resting*, and updates the knowledge base with perceived and latest context and situations. Then, Analyzer and Planner components are responsible for identifying possible requirement violations and generating an adaptation strategy, respectively, with the updated knowledge, while Executor will enact adaptation actions at runtime. These activities together are known as the MAPE-K feedback loop, which, while closely related to *environmental-perception*, is beyond the scope of this work. Interested reader can refer to [25], [26] for more details on the underlying self-adaptation mechanisms.

## V. CASE STUDY

In this section, we adopt a typical simplified smart room environment structured according to the proposed reference architecture. Multiple systems are often deployed in the smart room environment. Examples include the smart room system to maintain user comfort and the intelligent medical system to care for user health. The sensors in the smart room will perceive a wide variety of data, some of which will be needed and shared by both systems. For this study, we will illustrate (1) how *environmental-perception* as a separate first-class entity perceives and manages smart room information, (2) how the CPSs obtain their required information from the reference architecture and present adaptation behaviors, (3) smart room information reusability among multiple CPSs, and (4) the reusability for a system to be deployed in different application environments without modifications.

### A. SMART ROOM ENVIRONMENT

Facilitator Tier. In a smart room environment, there are a variety of raw data types such as temperature, humidity, and relative distance of users, as shown in the Figure 10. These values are directly perceived by the sensors from the facilitator tier: the air conditioner and humidifier are both off (set as 0) while the continuous values of distances from deployed sensors to the user is 1.3, 1.4 and 1.0 meter, respectively.

Validation Tier. This tier verifies the raw data through a predefined validation function to determine whether the sensor is damaged and whether the measured values are normal. The following formula describes a validation function with historical data and confidence intervals [27]:

$$vld(x) = \begin{cases} True & \bar{X} - z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \leq x \leq \bar{X} + z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \\ False & \text{Otherwise} \end{cases} \quad (1)$$

where $\bar{X}$ is the average of the $n$ historical data; $z_{\frac{\alpha}{2}}$ is the predefined standard score and $\frac{\sigma}{\sqrt{n}}$ is the standard error. Based on historical data, a hypothesis is made about the distribution of the current value. The confidence interval shows the upper and lower limits of which $x$ would fall inside with confidence level $\alpha$. When $x$ falls outside the interval, it is considered a small probability event, and the value is rejected according to the principle of the practical impossibility of small probability. Through this validation function, the value of a continuous variable, such as temperature, can be judged.

Pretreatment Tier. The pretreatment tier aggregates raw data into coarse-grained data items upon which prediction is made, then stores data items for easy access to the system. For example, relative humidity [28] can be obtained by

| Environmental Perception | | | | | | |
|---|---|---|---|---|---|---|
| Smart Room System | Level III | State of Interests | Sleep, Coma, Awake, Healthy, Sick, Critical | Situation | Slept, Healthy | |
| | Level II | Context Configuration | Heart_Rate, User_Temperature, User_Location | | | |
| Intelligent Medical System | Level III | State of Interests | Comfort_Level_low Comfort_Level_High, Comfort_Level_Average | Situation | Comfort_Level_Average | |
| | Level II | Context Configuration | Temperature, Relative_Humidity, Air_Conditioner, Humidifier, User_Location | | | |
| | Level I | Data Item | Temperature, Relative_Humidity, Air_Conditioner, Humidifier, Heart_Rate, User_Temperature, User_Location | Predicted Value | 30°C, 16.5%, off, off, 80bpm, 36.7°C, "on_chair" | |
| | | | | Current Value aggregated from raw values | 30°C, 16.6%, off, off, 80bpm, 36.6°C, "on_chair" | |
| | | Sensor & Raw Value | $Sen_{temperature}$ (30°C), $Sen_{absolute\_humidity}$ (800 gm/m$^3$), $Sen_{air\_conditioning}$ (off), $Sen_{humidifier}$ (off), $Sen_{heart\_rate}$ (80bpm), $Sen_{user\_temperature}$ (36.6), $Sen_{user\_distance\_1}$ (1.3m), $Sen_{user\_distance\_2}$ (1.4m), $Sen_{user\_distance\_3}$ (1.0m), | | | |

**FIGURE 10.** Example of ***Environmental-Perception*** **in a Smart Room.**

aggregating two raw data:

$$agg(x_1, x_2) = \frac{x_1}{f(x_2)} \times 100\% \tag{2}$$

where $x_1$ is the absolute humidity, $x_2$ the temperature; the function $f(x_2)$ calculates the maximum absolute humidity at $x_2$. Thus, relative humidity is the ratio of absolute humidity to the maximum absolute humidity that can be reached. The user's position can also be uniquely determined by three different sensors in the room and calculated whether he is "on the chair", "on the bed", or "others" according to the position of objects in this environment and aggregation function defined in advance.

In addition, the data items in the environment can be predicted to further understand environmental information. The following prediction function describes a moving average method:

$$pre(x_{t+1}) = \frac{1}{k}(x_{t-k+1} + x_{t-k+2}+, \ldots, x_{t-1} + x_t) \tag{3}$$

which is a method for predicting one future period with an average of $k$ recent actual captured values. For example, the next value might be 16.5% by applying this function to relative humidity. All data items will be stored for easy access to the systems.

### *B. CPSs IN THE SMART ROOM ENVIRONMENT*

*Smart room system.* Smart room systems are widely adopted, with the main function being to evaluate the habitability of the house by obtaining the environmental information in the room, and then makes adjustments that affect the environment [29], [30].

- Transport Tier. For the context configuration of the smart room system, Monitor will collect data items, such as temperature, relative humidity, and the user's location, delivered from the *environmental-perception* entity, and stores them in the knowledge base.
- Application Tier. In the smart room system, as shown in Figure 11, a situational decision tree (i.e., *idt* function where a non-leaf node denotes the context input while a leaf node represents the situation output) is constructed with the prior knowledge of experts. All data items in its context are abstracted into a situation where the comfort level is either average, low, or high. The situation in the current context – relative humidity 16.6%, humidifier off, temperature 30°C and air conditioner off – is "comfort_level_average" for the example in Figure 10. Situational models could also be obtained via clustering and learning user's behaviors and feedback.

System Analyzer will infer whether the goals are met based on the context and situations of the system, such as relative humidity whose predicted value will fall to 16.5%. Analyzer takes that information as the input of the reasoning engine and concludes that user goals can not be satisfied in a short time. Then, Planner matches the situation with the precondition of adaptation rules in the knowledge base and plans a series of adaptation behaviors, (e.g., turning on the humidifier and air conditioner in this case), which are carried out by Executor.

*Intelligent Medical System.* Another widely used application in the room environment is the intelligent medical system that determines when to remind patients to take medicine or call for emergency services by collecting symptoms-related data of human [31]–[33].

- Transport Tier. The context perceived by Monitor includes data items of the user's heart rate, temperature, and location.

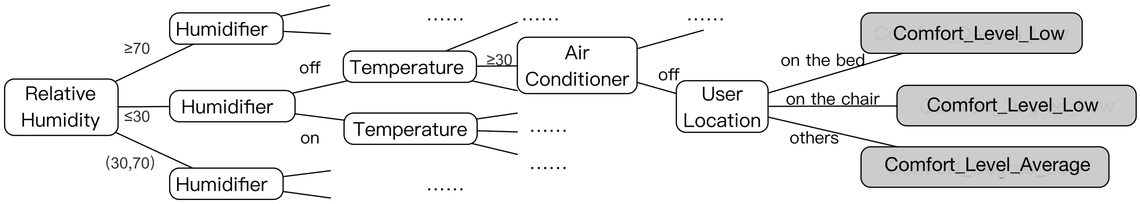



**FIGURE 11.** **Example of *Situation Identification* Function *idt* in a Smart Room System.**

- Application Tier. From the perspective of the intelligent medical system, all possible states of interest include user "asleep," "coma," "awake," "healthy," "sick," and "critical." With the context - the user in bed, the heart rate 80 bpm, the body temperature 36.6°C, and the predicted value 36.7°C - the situation identified in this application tier is user "asleep" and "healthy".

For the current context and situations in the intelligent medical system, user goals are met and there is no need to perform adaptation behaviors right away. In addition, to realize the proactive adaptation behaviors, the system has to predict requirement violations and take measures in advance. For instance, although the present user's heart rate and body temperature are within the normal range, and an identified situation is user "healthy", the prediction of a fluctuating heart rate would result in a reminding message for the user to take their medicine.

Based on the illustration of a simplified smart room environment, we have demonstrated that the *environmental-perception* can be divided into environment-independent data capture, system-related context awareness, and goal-oriented situation identification. This can be realized by the five-tier reference architecture proposed in the previous section.

For a specific environment, *environmental-perception*, as a separate first-class entity, independently perceives and processes all kinds of underlying information, shielding the CPSs from tedious processing details. It is convenient not only for a CPS to access and use the environmental information, but also for multiple CPSs deployed in that environment to reuse the environmental information. For instance, the user's location is required by both the smart room system and the medical system. When a new system is deployed, the environmental information is available to it by clearly defining its context configuration.

A system does not need to consider how to deal with complex and trivial fine-grained information; instead, it obtains the coarse-grained context using the context configuration. For example, the smart room system only needs to know whether the user is "on the chair", without considering how to deploy sensors and how to calculate the user's location using the aggregation function. When the deployment environment (i.e., room) of the system is updated, the number of sensors, the objects in the room and the placement of the items like chairs may be different from the previous environment. However, these differences are perceived and processed by the new *environmental-perception* entity. Through the five-tier implementation architecture, a system could easily access the user's location from the entity by context configuration. The core concern for the system is to understand this information, that is, identify the interesting situations. Based on that, a system can be applied to various application environments and make adaptation decisions to continuously meet system goals.

## VI. EMPIRICAL EVALUATION

This empirical evaluation aims to better understand the motivations and the benefits of the proposed *environmental-perception* and reference architecture from a practical perspective. We surveyed software engineers in system development with various backgrounds. We prepared an online questionnaire to collect data from participants. Then we invited 100 participants for the study through *email* and *LinkedIn network* and **32** participants responded. After conducting pre-testing and introducing them to this work, i.e., three levels of environmental-perception as an independent entity and the reference architecture for information collection serving as a medium between environment and various CPSs, they were asked to fill in the questionnaire. The estimated time to complete each survey was around 15 minutes comprising a 10-minute introduction and a 5-minute section with 8 questions:

Q1. Which field does the system you are involved in belong to?
Q2. How does your system perceive the environment?
Q3. How many systems are in the environment where your system is deployed?
Q4. What is the overlap of environmental information between systems if there is more than one deployed?
Q5. Can the understanding of the environmental information for a system be promoted by explicitly differentiating the definition of environment, context, and situation?
Q6. Is it necessary for *environmental-perception* to be a separate first-class entity?
Q7. With the proposed reference architecture, can better reuse of environmental information among different systems be achieved?
Q8. With the proposed reference architecture, can a system be more reusable to different deployment environments?

The first four questions investigate the current status of the environment and environmental information perception

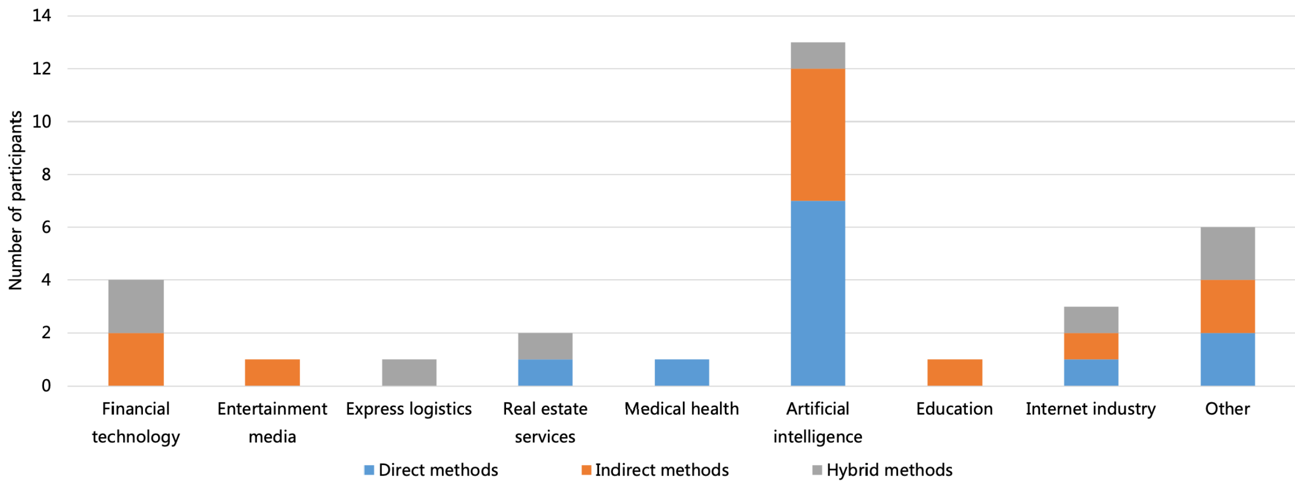



**FIGURE 12. Field Distribution of Respondents and Their Systems' Methods of Perceiving the Environmental Information.**

from the industry perspectives, further supporting the motives of our work. In Q3 and Q4, we are studying whether it is necessary to treat *environmental-perception* as a first-class entity with three levels. In the last two questions, we are investigating the industrial comments on the proposed reference architecture for *Environmental-Perception*.

### A. RESULTS

#### 1) ANALYSIS OF THE SURVEY

32 respondents are from more than nine areas as shown in Figure 12, with 13 from artificial intelligence (abbr. AI), 4 from financial technology, 3 from the internet industry. Regarding the methods of perceiving the environment, three options are given, including direct methods (e.g., deploying sensors to capture environmental information), indirect methods (e.g., communicating with other systems to obtain information), and hybrid methods that combine the first two methods. A total of 12 respondents (37.5%) answered direct methods, such as in the medical health system, whose environmental information is not available otherwise due to the specificity of the required information. However, in the field of fintech, for example, systems make more or less indirect use of third-party data, which means that at least some of the data can be reused across multiple systems.

In response to the number of systems in the environment where the respondent's system was deployed, as shown in Figure 13, exactly half of the responses were more than one, with six responding with 2 systems, one responding with 3, and nine responding with 3 or more. Furthermore, in the cases where there were multiple systems in an environment, 34 percent indicated that there was some overlap between the required environmental information while 16 percent believed there was a substantial overlap. The above results demonstrate that it is not uncommon in industry for multiple systems to be deployed in the same environment, and that there was indeed environmental information overlap between systems. The overlaped information can be shared by communication and collaboration.

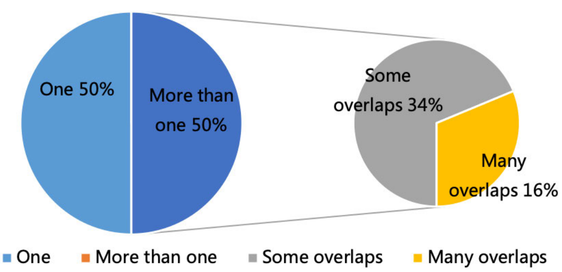



**FIGURE 13. The number of Systems in an Environment and the Information Overlap between Them.**

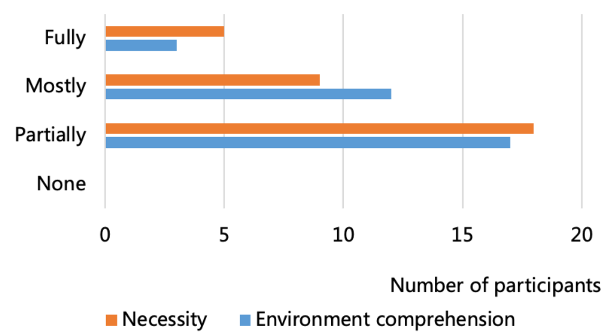



**FIGURE 14. The understanding of environment by explicitly differentiating the definitions and the necessity of *environmental-perception* as a separate first class entity.**

With respect to Q5 and Q6, when asked whether the environmental information can be better comprehended by explicitly separating the definition of environment, context, and situation, all of them believed that these concepts are constructive and that it is also necessary to embrace the *environmental-perception* as a first-class realizable entity, albeit to varying degrees of agreement, as shown in Figure 14. The results in Figure 15 illustrates the reusability of the information in a certain environment for multiple deployed systems, and the reusability of a specific system applied to various application environments without the modification on

| | Options | Quantity | Percentage |
|---|---|---|---|
| Reusability on Environmental Data | Non-reusable | 3 | 9.38% |
| | Part of data reusable | 17 | 53.13% |
| | Most data reusable | 10 | 31.25% |
| | Fully reusable | 2 | 6.25% |
| Total | | 32 | |
| Reusability for a System | Non-reusable | 3 | 9.38% |
| | Reusable in some environments | 16 | 50.00% |
| | Reusable in most environments | 10 | 31.25% |
| | Fully reusable | 3 | 9.38% |
| Total | | 32 | |

**FIGURE 15.** **The reusability for the environmental information and for a system deployed in various environment by adopting our proposed reference architecture.**

the mechanism for information perception. Three of them from AI considered it unreusable. The reasons they gave include very different types of data such as text, audio and video, and various ways of data processing like denoising and principal component analysis. This diversity leads to the unique data collection in AI experiments, but by the respondents' own admission this is a very primitive way and further improvement of data transferability are needed. In addition, a large majority answered that our reference architecture enables reusability to varying degrees.

### B. DISCUSSION

The results from our industrial application survey emphasize the motivations and benefits of this work. In summary, the survey results have shown that: (i) usually more than one system is deployed in an environment, (ii) there is indeed overlap of required environmental information between systems, and (iii) there are both indirect and hybrid methods of perceiving environmental information. These facts prompt us to view *environmental-perception* as a separate first-class entity, like a third party providing data on demand. However some respondents questioned whether the *environmental-perception* entity might need to collect all data in an environment, which might be innumerable and impossible, to satisfy all potential deployed systems. We argue that as more systems are deployed and new information is required, this entity does need to deploy new sensors to detect and aggregate the information in need. Besides, some of the data might be quite unusual or private and should not be disclosed to other systems due to the particularity of some industries, it would be inappropriate for *environmental-perception* entity to collect this information and transmit it in these cases. However, even if only a portion of the data can be reused, the three levels of perception and reference architecture do make sense, which is also agreed by the questioners.

We acknowledge that the boundaries of question options, such as between some overlaps and many overlaps or between partially agreement and mostly agreement, are difficult to quantify and that each individual may have different criteria. Though our survey is preliminary and there is no quantification on the degree of reusability, comprehension, and necessity, etc, the analysis results indicate the benefits of enhancing environmental information comprehension with explicit concept separation and information reusability with an independent *environmental-perception* entity with the proposed reference architecture. Moreover, to practically integrate the conceptual model to real applications and verify the validity of the proposed reference architecture, industrial standards such as Technology Acceptance Model (abbr. TAM) with further quantification of Perceived Usefulness (abbr. PU) and Perceived Ease-of-Use (abbr. PEOU) [34], [35] is necessary to be adopted in practical applications. While this is an area for future work, the widely recognized ISO model to which our proposed reference architecture refers backs *Environmental-Perception* in PU and PEOU to some extent. In addition, our reference architecture is based on the sound theoretical framework defined by the reasonable three layers, featuring theoretical and conceptual feasibility and validity.

## VII. RELATED WORK

Existing technology extensively relies on robust CPSs to achieve specific goals and adapt to uncertainties [3]. Much of the existing work has focused on analysis and planning activities to analyze the available actions and their potential outcome on system goals, and to plan corresponding adaptation decisions, assuming adequate monitoring in place [1], [2]. Perceiving environmental information as the prerequisite for a CPS to be reliable and adaptive, however, has not gained the deserved attention. Some closely related work on information perception are summarized in Table 1.

The environment was pioneered as a first-class entity in the multi-agent system instead of being dealt with implicitly or in an ad hoc manner by Weyns *et al.* [8]. Their work fundamentally changes our understanding of the environment. In our work, we further consider the process of *environmental-perception*, rather than the environment itself, as a separate first-class entity, thereby unbinding the system from the environment and providing two types of reusability. Ferber *et al.* [36] integrate the physical and social environment with organization into the Agent-Group-Role(AGR) model by introducing relations between world, spaces, areas, groups, modes, bodies, and roles at the concrete level. Instead, we consider three vertical levels, Data-Context-Situation, to support various CPSs deployed in an environment. Jiang *et al.* [29] implement *environmental-perception* as a four-tier structure with the service tier, the presentation tier, the management tier, and the resource tier to facilitate the access of environmental information, providing a mapping mechanism between the environment elements recognized by the system and the real environment. However, their work does not consider, as we do in this work, how to abstract, organize, manage the environmental information according to the needs of various systems, or support information reuse. Meng *et al.* [37] argue information and emergency can be handled by coordination and information sharing among sensing and executive agents so that real-time data can be delivered to the control center to assist in decision making. Although there is an emphasis on information sharing among different components in a system, the direct perception method of the system neither supports information reuse among various systems, nor provides perception implementation reuse among different deployment environments.

In the definition of the context, Li *et al.* [38] interpret context as a set of environmental elements related to a specific action; the context is derived from action-related constraints at runtime and managed by an active context manager. This is similar to our definition as a set of aggregated data in the deployed environment acquired by the system. Park *et al.* [39] consider context information as services, sensors, environments, users, and tasks, which is modeled at construction time. A context-aware framework is mainly used to manage and coordinate context and to establish the basis for interoperability among entities. A service-oriented context-aware middleware is developed by Hafiddi *et al.* [40], providing service information through three modules: the context manager, the event notification, and the task engine. Li *et al.* [41] propose a general conceptual model of the system with four parts: context awareness, context evolution, agent and application interface. Sridevi *et al.* [42] describe a context-aware framework based on asynchronous communication between the perception layer and application layer through publish-subscribe mode. These frameworks define how to collect the contextual information from the environment using sensors in the design phase. However, they do not show a partitioning of the de facto different layers, nor do they decouple functions at different layers, resulting in the inability to reuse environmental information for multiple systems.

The situation tends to give the system context additional information on the basis of system intentions.

**TABLE 1. A taxonomy of the research papers in environmental-perception.**

| Reference | Content | Contributions | Compared to our work |
|---|---|---|---|
| **Weyns et al. [8]** | Pioneer the environment as a first-class entity | Fundamentally change our understanding of the environment | We consider the process of *environmental-perception* as a separate first-class entity unbinding the system from its deployed environment and providing two types of reusability |
| **Ferber et al. [36]** | Integrate the physical and social environment with organization into the Agent-Group-Role(AGR) model by | Provide different levels (e.g., agent, group, world) of environmental information | We consider three vertical levels Data-Context-Situation to support various CPSs deployed in an environment |
| **Jiang et al. [29]** | Implement *environmental perception* as a four-tier structure with service tier, presentation tier, management tier, and resource tier | Facilitate the access of environmental information | Their work does not consider how to abstract, organize or manage the environmental information according to the needs of various systems nor support information reuse |
| **Meng et al. [37]** | Envision information and emergency handling through coordination and information sharing among sensing and executive components | Emphasize information sharing among different components for information perception and emergency handling | They do not support information reuse among various systems, or provide perception implementation reuse among different deployment environments. |
| **Li et al. [38]** | Derive information from action-related constraints at runtime and manage it by an active context manager | Define context as a set of environmental elements related to a specific action | Their context definition is similar to ours as a set of aggregated data in the deployed environment acquired by the system |
| **Viroli et al. [39]** | Consider the information as services, sensors, environments, users, and tasks, modeled at construction time | Emphasize context-awareness to manage and coordinate context | They do not show a partitioning of the de facto different layers, nor do they decouple the different functions to be implemented at different layers, resulting in the inability to reuse environmental information for multiple systems. |
| **Hadiddi et al. [40]** | Provide service-related information through three modules: context manager, event notification and task engine | | |
| **Li et al. [41]** | Propose a general conceptual model of the system with four parts: context awareness, context evolution, agent and application interface. | | |
| **Sridevi et al. [42]** | Describe a context-aware framework based on asynchronous communication between the perception layer and application layer through publish-subscribe mode. | | |
| **Carl Chang [43]** | Define the situation as a tuple with object, intention, and context. | Propose a situation-aware framework | They confuse the situation with the context, and do not explicitly link the situation to the objectives of the system. |
| **Fredericks et al. [44]** | Specify the situation in terms of context parameters and dynamically identify the distinct situation at run-time to discover optimal system configurations | | |

Chang [43] proposes a situation-aware framework, defining the situation as a tuple with an object, intention, and context. Fredericks *et al.* [44] specify the situation in terms of context parameters and dynamically identify the distinct situation at run-time to discover optimal system configurations. These, in turn, confuse the situation with the context, and do not explicitly link the situation to system objectives. On the contrary, we explicitly define the situation as an interesting state, integrating the intention and goal of the system through the identification function. This is similar to Semantic Web (SW) [45] in pervasive computing, aiming to formally interpret intended semantics from heterogeneous sources to support the management of information. Additionally, Ye *et al.* [46] further introduce semantic relations at different levels of abstraction and reason in the presence of extensive uncertainty. In general, the context and situation have not been explored much in the field of cyber-physical systems. Context-awareness and situation-identification should also be the core functions of the cyber-physical systems, like the pervasive system, to further enhance the system's ability to perceive, adapt, and transform the environment.

## VIII. CONCLUSION

Cyber-physical systems have become increasingly important especially in the environment with the integration of human society, the physical world, and the cyber world. Though many excellent research efforts have been put into this area, *environmental perception* as a field is still in its infancy, and existing approaches are not adequate to address today's ever-changing and ever-expanding various environments. Following Weyns' proposal in the multi-agent system that *environment-perception* should be a first-class entity as an essential part of the system, we further propose *environmental-perception* as a first-class entity existing independently and define data items, context, and situation. Subsequently, we describe a five-tier reference architecture with the borrowed thoughts from ISO protocol. In this framework, data capture from the facilitator to transport tier provides a mechanism for perceiving the dynamics and evolution of the environments, facilitating environmental information reusability. Meanwhile, transport and application tiers closely linked to the system enable it to have the abilities of context awareness and situation identification. This ensures the system goals are continuously satisfied by having it performing adaptation behaviors. In the end, to illustrate the applicability of the perception framework, we present an example from a smart room environment. A survey is also included to indicate industrial motivations and benefits of the *environmental perception* as a first-class entity.

In our future research, we plan to apply the reference architecture to more practical scenarios to strengthen its applicability, such as the evocative examples of *smart buildings* and *smart cities*. We also aim to refine the reference architecture by applying/instantiating advanced AI algorithms to obtain specific methods of data prediction or situation identification. In addition, we will focus on designing a granular quantitative questionnaire to further understand the requirements of *environmental-perception* from the industry.

## ACKNOWLEDGMENT

The authors would like to thank John Sun for polishing and improving the clarity of the draft and Gabriel A. Tauber for proofreading a nearly final version of this article. They also thank the reviewers for their valuable comments.

## REFERENCES


[1] C. Tsigkanos, L. Pasquale, C. Ghezzi, and B. Nuseibeh, "On the interplay between cyber and physical spaces for adaptive security," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 3, pp. 466–480, May 2018, doi: 10.1109/TDSC.2016.2599880.

[2] N. Li, D. Bai, Y. Peng, Z. Yang, and W. Jiao, "Verifying stochastic behaviors of decentralized self-adaptive systems: A formal modeling and simulation based approach," in *Proc. IEEE Int. Conf. Softw. Qual., Rel. Secur. (QRS)*, Lisbon, Portugal, Jul. 2018, pp. 67–74, doi: 10.1109/QRS.2018.00020.

[3] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. 11th IEEE Int. Symp. Object Component-Oriented Real-Time Distrib. Comput. (ISORC)*, Orlando, FL, USA, May 2008, pp. 363–369, doi: 10.1109/ISORC.2008.25.

[4] B. H. C. Cheng *et al.*, "Software engineering for self-adaptive systems: A research roadmap," in *Software Engineering for Self-Adaptive Systems*. Berlin, Germany: Springer, 2009, pp. 1–26, doi: 10.1007/978-3-642-02161-9_1.

[5] R. D. Lemos *et al.*, "Software engineering for self-adaptive systems: A second research roadmap," in *Software Engineering for Self-Adaptive Systems II*. Berlin, Germany: Springer, 2010, pp. 1–32, doi: 10.1007/978-3-642-35813-5_1.

[6] S. K. Damodaran and P. D. Rowe, "Limitations on observability of effects in cyber-physical systems," in *Proc. 6th Annu. Symp. Hot Topics Sci. Secur. (HotSoS)*, Nashville, TN, USA, 2019, pp. 1–10, doi: 10.1145/3314058.3314065.

[7] H. Kopetz, *Simplicity is Complex: Foundations of Cyber-Physical System Design*. New York, NY, USA: Springer, 2019, doi: 10.1007/978-3-030-20411-2.

[8] D. Weyns, A. Omicini, and J. Odell, "Environment as a first class abstraction in multiagent systems," *Auto. Agents Multi-Agent Syst.*, vol. 14, no. 1, pp. 5–30, 2007, doi: 10.1007/s10458-006-0012-0.

[9] D. Weyns, H. V. D. Parunak, F. Michel, T. Holvoet, and J. Ferber, "Environments for multiagent systems state-of-the-art and research challenges," in *Proc. 1st Int. Workshop Environ. Multi-Agent Syst.*, New York, NY, USA, Jul. 2004, pp. 1–47, doi: 10.1007/978-3-540-32259-7_1.

[10] S. VanSyckel and C. Becker, "A survey of proactive pervasive computing," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. Adjunct Publication-UbiComp Adjunct*, Seattle, WA, USA, 2014, pp. 421–430, doi: 10.1145/2638728.2641672.

[11] D. Tennenhouse, "Proactive computing," *Commun. ACM*, vol. 43, no. 5, pp. 43–50, May 2000.

[12] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts Essentials*, 2nd ed. Hoboken, NJ, USA: Wiley, 2014.

[13] P. Brézillon and A. J. Gonzalez, Eds., *Context in Computing: A Cross-Disciplinary Approach for Modeling the Real World*. New York, NY, USA: Springer, 2014, doi: 10.1007/978-1-4939-1887-4.

[14] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and Ubiquitous Computing*. Berlin, Germany: Springer, 1999, pp. 304–307, doi: 10.1007/3-540-48157-5_29.

[15] "The design and implementation of self-adaptive service composition system based on context awareness," M.S. thesis, School Electron. Inf. Elect. Eng., Shanghai Jiao Tong Univ., Shanghai, China, 2016.

[16] C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker, "A survey on engineering approaches for self-adaptive systems," *Pervasive Mobile Comput.*, vol. 17, pp. 184–206, Feb. 2015, doi: 10.1016/j.pmcj.2014.09.009.

[17] J. Ye, S. Dobson, and S. McKeever, "Situation identification techniques in pervasive computing: A review," *Pervas. Mobile Comput.*, vol. 8, no. 1, pp. 36–66, Feb. 2012, doi: 10.1016/j.pmcj.2011.01.004.

[18] G. G. McBride, "ISO/OSI and TCP/IP network model characteristics," in *Information Security Management Handbook, Fifth Edition*, 6th ed. Boca Raton, FL, USA: CRC Press, 2007, pp. 1917–1928, doi: 10.1201/9781439833032.ch144.

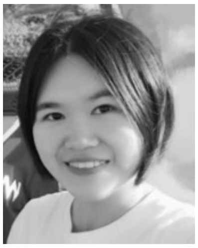
[19] (2020). *Prototypical Reference Architecture And Partial Implementation on the Smart Room Case Study*. [Online]. Available: https://github.com/sherry1912/EnvironmentalPerception

[20] B. Ding, H.-M. Wang, and D.-X. Shi, "Constructing software with self-adaptability," *J. Softw.*, vol. 24, no. 9, pp. 1981–2000, 2013.

[21] L. Sakizloglou, S. Ghahremani, T. Brand, M. Barkowsky, and H. Giese, "Towards highly scalable runtime models with history," in *Proc. IEEE/ACM 15th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst. (SEAMS)*, Seoul, South Korea, Jun./Jul. 2020, pp. 188–194, doi: 10.1145/3387939.3388614.

[22] R. Mayrhofer, H. Radi, and A. Ferscha, "Recognizing and predicting context by learning from user behavior," *Radiomatics, J. Commun. Eng., Special Issue Adv. Mobile Multimedia*, vol. 1, no. 1, pp. 25–35, 2004.

[23] D. Ameyed, M. Miraoui, and C. Tadj, "A survey of prediction approach in pervasive computing," *Int. J. Sci. Eng. Res.*, vol. 6, no. 5, pp. 306–316, 2015.
[24] Q. Li, Y. Tian, Q. Wu, Q. Cao, H. Shen, and H. Long, "A cloud-fog-edge closed-loop feedback security risk prediction method," *IEEE Access*, vol. 8, pp. 29004–29020, 2020, doi: 10.1109/ACCESS.2020.2972032.
[25] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003, doi: 10.1109/MC.2003.1160055.
[26] D. Weyns, B. R. Schmerl, V. Grassi, S. Malek, R. Mirandola, C. Prehofer, J. Wuttke, J. Andersson, H. Giese, and K. M. Göschka, "On patterns for decentralized control in self-adaptive systems," in *Software Engineering for Self-Adaptive Systems II*. Berlin, Germany: Springer, 2010, pp. 76–107, doi: 10.1007/978-3-642-35813-5_4.
[27] *Confidence Interval*. Accessed: May 2015. [Online]. Available: https://en.wikipedia.org/wiki/Confidence_interval
[28] *How to Find Relative Humidity*. Accessed: Aug. 2017. [Online]. Available: https://sciencing.com/relative-humidity-7611453.html
[29] H. Jiang, Y. Liu, H. Lin, and W. Jiao, "Environment model of self-adaptive software and its implementation," *J. Frontiers Comput. Sci. Technol.*, vol. 10, no. 3, pp. 301–310, 2016.
[30] C. Koehler, B. D. Ziebart, J. Mankoff, and A. K. Dey, "TherML: Occupancy prediction for thermostat control," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, Zürich, Switzerland, 2013, pp. 103–112, doi: 10.1145/2493432.2493441.
[31] J. Grosinger, F. Pecora, and A. Saffiotti, "Making robots proactive through equilibrium maintenance," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, NY, USA, 2016, pp. 3375–3381.
[32] D. Weyns and M. U. Iftikhar, "Model-based simulation at runtime for self-adaptive systems," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Wuerzburg, Germany, Jul. 2016, pp. 364–373, doi: 10.1109/ICAC.2016.67.
[33] A. Machado, V. Maran, I. Augustin, L. K. Wives, and J. P. M. de Oliveira, "Reactive, proactive, and extensible situation-awareness in ambient assisted living," *Expert Syst. Appl.*, vol. 76, pp. 21–35, Jun. 2017, doi: 10.1016/j.eswa.2017.01.033.
[34] I. Aburagaga, M. Agoyi, and I. Elgedawy, "Assessing faculty's use of social network tools in Libyan higher education via a technology acceptance model," *IEEE Access*, vol. 8, pp. 116415–116430, 2020, doi: 10.1109/ACCESS.2020.3004200.
[35] C. Fan, "Applied the technology acceptance model to survey the mobile-learning adoption behavior in science museum," in *Cross-Cultural Design. Applications in Health, Learning, Communication, and Creativity*. Cham, Switzerland: Springer, 2020, pp. 270–280, doi: 10.1007/978-3-030-49913-6_23.
[36] J. Ferber, F. Michel, and J. Báez-Barranco, "AGRE: Integrating environments with organizations," in *Environments for Multi-Agent Systems*. Berlin, Germany: Springer, 2004, pp. 48–56, doi: 10.1007/978-3-540-32259-7_2.
[37] W. Meng, Q. Liu, W. Xu, and Z. Zhou, "A cyber-physical system for public environment perception and emergency handling," in *Proc. IEEE Int. Conf. High Perform. Comput. Commun.*, Banff, AB, Canada, Sep. 2011, pp. 734–738, doi: 10.1109/HPCC.2011.104.
[38] X. Li, X. Tao, J. Lu, and W. Song, "Specification and runtime verification for activity-oriented context-aware applications," *Ruan Jian Xue Bao/J. Softw.*, vol. 28, no. 5, pp. 1167–1182, 2017.
[39] N.-S. Park, K.-W. Lee, and H. Kim, "A middleware for supporting context-aware services in mobile and ubiquitous environment," in *Proc. Int. Conf. Mobile Bus. (ICMB)*, 2005, pp. 694–697.
[40] H. Hafiddi, H. Baidouri, M. Nassar, and A. Kriouile, "An aspect based pattern for context-awareness of services," *Int. J. Comput. Sci. Netw. Secur.*, vol. 12, no. 1, pp. 71–78, 2012.
[41] R. Li and R. Li, "A survey of context-aware computing and system framework," *J. Comput. Res. Develop.*, vol. 2, pp. 269–276, 2007.
[42] S. Sridevi and S. R. Bhattacharya Pitchiah, "Context aware framework," in *Proc. MobiQuitous*, 2010, pp. 358–363.
[43] C. K. Chang, "Situation analytics: A foundation for a new software engineering paradigm," *Computer*, vol. 49, no. 1, pp. 24–33, Jan. 2016.
[44] E. M. Fredericks, I. Gerostathopoulos, C. Krupitzer, and T. Vogel, "Planning as optimization: Dynamically discovering optimal configurations for runtime situations," in *Proc. IEEE 13th Int. Conf. Self-Adaptive Self-Organizing Syst. (SASO)*, Umea, Sweden, Jun. 2019, pp. 1–10, doi: 10.1109/SASO.2019.00010.
[45] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic Web revisited," *IEEE Intell. Syst.*, vol. 21, no. 3, pp. 96–101, May 2006, doi: 10.1109/MIS.2006.62.
[46] J. Ye, S. Dasiopoulou, G. Stevenson, G. Meditskos, E. Kontopoulos, I. Kompatsiaris, and S. Dobson, "Semantic Web technologies in pervasive computing: A survey and research roadmap," *Pervasive Mobile Comput.*, vol. 23, pp. 1–25, Oct. 2015, doi: 10.1016/j.pmcj.2014.12.009.



**NIANYU LI** received the B.Sc. degree in software engineering from the Nanjing University of Aeronautics and Astronautics. She is currently pursuing the Ph.D. degree in computer software and theory with Peking University under the supervision of Prof. Zhi Jin and Prof. Wenpin Jiao. She is a Visiting Research Student with the National Institute of Informatics (NII), Japan, and Carnegie Mellon University (CMU), USA. Her research interests include (human-involved) self-adaptive systems, cyber-physical systems, modeling, and formal verification.

**KUN LIU** received the B.Sc. degree in software engineering from the Nanjing University of Aeronautics and Astronautics. He is currently pursuing the Ph.D. degree in computer software and theory with Peking University under the supervision of Prof. Wenpin Jiao. He was a Visiting Research Student with the National Institute of Informatics (NII), Japan, from September 2019 to January 2020. His research interests include self-adaptive systems and multiagent systems (especially multiagent reinforcement learning).

**ZHENGYIN CHEN** received the B.Sc. degree in computer science and technology from Peking University, where he is currently pursuing the Ph.D. degree in computer software and theory under the supervision of Prof. Wenpin Jiao. His research interests include self-adaptive systems, proactive software systems, and context prediction.

**WENPIN JIAO** received the B.S. and M.S. degrees in computer science from the East China University of Science and Technology, in 1991 and 1997, respectively, and the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, in 2000. He is currently a Full Professor with the School of Electronics Engineering and Computer Science, Peking University. His research interests include software engineering, multiagent systems, and intelligent software technology.