Received: 23 November 2017 | Revised: 6 February 2018 | Accepted: 26 February 2018
DOI: 10.1002/cpe.4481

WILEY

SPECIAL ISSUE PAPER

# Reliability, failure detection and prevention in cyber-physical systems (CPSs) with agents

Teodora Sanislav[1] | Sherali Zeadally[2] | George Dan Mois[1] | Hacène Fouchal[3]

[1] Technical University of Cluj-Napoca, Cluj-Napoca, Romania
[2] University of Kentucky, Lexington, KY, USA
[3] Université de Reims Champagne-Ardenne, Reims, France

**Correspondence**
Teodora Sanislav, Technical University of Cluj-Napoca, Cluj-Napoca, Romania.
Email: teodora.sanislav@aut.utcluj.ro

**Funding information**
Technical University of Cluj-Napoca, Internal Competition CICDI-2017, Grant/Award Number: 1998/12.07.2017

## Summary

The complexity of cyber-physical systems (CPSs) opens up several challenges regarding the assurance of high levels of reliability. Therefore, we need to develop a new data quality analysis and new CPS components/subsystems for the detection, isolation, recovery, and prevention of failures and intrusion events. Furthermore, we have to develop new reliability assessment models and metrics in order to properly evaluate CPSs. Agent technologies, through their characteristics such as autonomy, social ability, reactivity, and proactivity, have the potential to address these challenges and requirements. We propose a multi-agent based solution, namely ReliaCPS, for detecting and preventing the failures of the system components of CPSs. We also use a case study of a CPS for monitoring ambient parameters to evaluate the performance of the proposed approach. Our results indicate improvements in terms of reliability metrics (32.69% for the mean time between failures and mean time to failure, 50% for the mean time to repair and 0.94% for reliability function over time).

KEYWORDS

agents, cyber-physical systems, failure detection, failure prevention, reliability

## 1 | INTRODUCTION

Information and communication technology (ICT) has evolved significantly over the past few decades and has become ubiquitous in so much of our daily lives. Every industry uses ICT to improve its processes and products. Cyber-physical systems (CPSs)[1-3] have emerged as a natural response to the increasing need for innovation in solving key challenges of our society such as mobility, the ageing population, limited resources, and climate change. CPSs are engineered systems that connect the real world through sensors and actuators with the virtual world of information processing.[4] This type of complex systems implies the coexistence of diverse components (ie, software systems, communications technologies, embedded hardware) that collaborate together to create a global behavior, exceeding the functionality and competitiveness of current systems used in sectors such as transportation, building design and automation, environment, energy, healthcare, agriculture, and manufacturing (Figure 1).

CPSs have the following defining characteristics[5]:

- They monitor and control physical and organizational processes.
- They adapt in response to the uncertainty of environments.
- They require a high degree of dependability (availability, reliability, safety, integrity, confidentiality, maintainability).
- They require hierarchical decision systems with a high degree of autonomy at different levels.
- They are large-scale, distributed, heterogeneous, and interconnected systems spanning different application domains.
- They involve significant user interactions.
- They integrate different engineering disciplines.

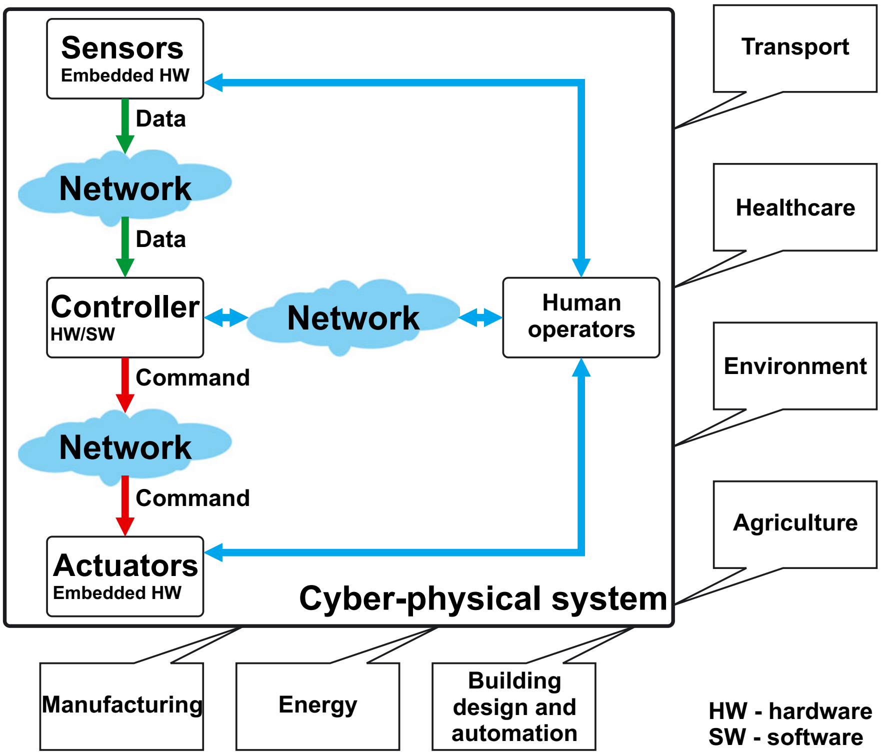


**FIGURE 1** Overview of the main characteristics of cyber-physical systems

These characteristics make CPSs suitable for deployment in various application domains. For example, in the transportation field, a CPS includes intelligent traffic control systems and autonomous vehicles capable of monitoring and controlling the safety and the mobility of passengers in public transport systems or of vehicles on road/rail networks. A CPS applied in the building design field can monitor and control heating, ventilation, and air conditioning (HVAC) as well as lighting in order to improve the comfort and safety in homes and offices. Also, a CPS can collect environmental data, generate warnings of environmental threats, and offer decision support to public authorities. CPSs used in the energy domain focus on power generation and distribution (smart grids) and energy conservation while taking into consideration resource savings. Systems capable of monitoring human health and supporting the elderly by detecting illnesses and accidents, using non-invasive wearable sensors, medical devices, and robots, represent CPSs for healthcare. In the agriculture domain, these large-scale systems can optimize crop yields and the water resource and reduce the use of chemicals. Intelligent and efficient production and management systems and manufacturing lines making use of robotic technologies are examples of CPSs in the field of manufacturing.

**Contributions of this work.**

All the CPSs examples above have to be high-confidence systems, and therefore, ensuring safety, security, scalability, and reliability are important requirements for them. The CPSs always need to function reliably and behave in a way the users expect them to. In this work, we focus on reliability assurance within CPSs. We summarize the main contributions of this work as follows:

- We present the state-of-the-art research results on reliability in CPSs from the following perspectives: data reliability, reliability assurance, and reliability analysis/evaluation.
- We discuss the current reliability techniques proposed for CPSs.
- Based on the literature review, we propose a multi-agent–based solution, called ReliaCPS, which can provide CPSs reliability with high assurance.
- We evaluate the proposed solution in terms of several reliability metrics (mean time between failures, mean time to failure, mean time to repair, and reliability over time) using a CPS case study.

In the remainder of this paper, we discuss reliability support within CPSs by considering recent related research efforts in several application domains and highlighting their approaches. Then, we propose an agent-based approach that improves the reliability of a CPS. Our approach includes two important reliability aspects: detection and prevention of failures. Next, we evaluate the proposed solution's performance. Finally, we make some concluding remarks.

## 2 | RELIABILITY IN CPSs

Dependability, survivability, and trustworthiness, which are essentially equivalent concepts, define the same property that ensures the protection of societal infrastructures monitored and controlled by CPSs against various threats. The dependability concept is the most comprehensive of these because it considers all classes of threats and their possible combinations. It comprises the following attributes: availability, reliability, safety, integrity, confidentiality, and maintainability, which can be more or less important, depending on the application (ie, availability is always required, while reliability, safety, or confidentiality may or may not be required).[6]

According to the work of Avizienis et al,[6] the reliability of a system is the "measure of the continuous delivery of a correct service or, equivalently, of the time to failure." In case of CPSs, which are safety-critical systems with heterogeneous components, reliability is highly required and is very important. This is why the challenge of reliability in CPSs is currently being addressed by the research community from various perspectives that include data reliability, reliability assurance, and reliability analysis/evaluation.

### 2.1 | Data reliability

Within a CPS, the data gathered from spatially distributed physical or organizational processes through sensors influences the decision making related to its control actions. The resulting decisions are materialized using actuators, and thus, the loop between the physical and cyber components is closed.[7] In the decision making stage, the CPS human operators are informed through real-time data and can intervene and modify the decisions which are then sent to the actuators, leading to human-in-the-loop (HIL) systems. Also, the data can be used to achieve other CPS tasks such as fault detection and localization, state estimation, and short/long-term forecasting.[8] Given the importance of the observed data, it must be reliable (accurate, accessible, timeliness and consistent). Abbas and his colleagues[8] highlighted that any malicious attack on sensors can have a severe impact on the physical processes and on the entire CPS. In this context, they formalized the notion of attack-resiliency, and they presented three approaches to achieve it: resilient sensor placement, resilient sensor network topology design, and resilient data aggregation algorithms design, along with some examples.

Wu[9] proposed a method for the automated online evaluation (AOE) of CPS data, which works in parallel with the CPS. His approach evaluates data quality of the system workflow and provides HIL feedback for improving the reliability. Data quality analyses such as machine learning, data mining, statistical analysis, and self-tuning techniques such as data classification, redundancy checking and trend detection are combined to increase the overall CPS's reliability.

### 2.2 | Reliability assurance

Being large-scale complex systems composed of heterogeneous computerized devices, CPSs are susceptible to a wide range of failures that can be broadly classified into two main groups, namely random component failures, caused by fault occurrences, and intentional failure of devices that can be caused by external attacks.[10] Since CPSs have high degrees of interconnectivity and are generally large-scale, they are particularly prone to cascading failures, a phenomenon in which the unexpected malfunction of a relative small number of sub-components triggers a sequence of failures affecting the entire system's operation. As CPSs rely heavily on information exchange, they are highly vulnerable to attacks that can cause certain communication nodes to fail and make other components malfunction because of increased additional communication and processing loads (usually denoted as denial-of-service attacks). By considering these factors, the reliability of a CPS can be ensured by efficient failure and intrusion detection, isolation, recovery, and prevention approaches.

Guo and his colleagues[11] developed a preliminary model for failure detection within a CPS for the manufacturing field. The model is based on the mailbox mechanism and sensors data fusion, and it is the foundation of a primary diagnosis system that has been tested to verify its delay and reliability. Unfortunately, its use is not deployed in a CPS.

Mitchell and Chen[12] developed an Intrusion Detection and Response System (IDRS) for detecting and responding to different attacks at runtime within a CPS with many sensors nodes. The proposed IDRS takes advantage of a host intrusion detection protocol, which permits the mapping of a specification into a state machine (good and bad states) and the measurement of a node's deviation from good states in order to detect intrusion. This system can be applied in several domains, such as environment, healthcare, and transportation.

Several research works describe CPSs as networks of small clusters of sub-networks, interconnected by communication links, fragile to cascading failures.[13,14] Furthermore, the literature includes many studies[15-18] that address this problem in complex systems, many of which cover the domain of power grids, which represent typical CPSs. In power grids, it has been proven that they have a higher rate of occurrences with effects such as electrical instability and large blackouts.[19-22] The work of Babalola et al[22] prevents cascading failures in the smart grid by leveraging a multi-agent system and an algorithm based on mathematical combinations to re-dispatch the power from the generators in case of multiple contingencies. The mathematical combinations used by the system's algorithm in case of fault occurrences are selected based on empirical historical economic dispatch data. Furthermore, cascading failures represent a major issue also in the case of other CPSs such as the ones in the transportation, utilities, and communication domains. Assuring high levels of reliability within a CPS is of great importance, and system architects, researchers, and engineers try to mitigate cascading failures through the implementation of self-adaptive and self-organizing techniques, hardware and software fault-tolerance, and others.

**TABLE 1** Definitions of the main CPS reliability metrics

| Reliability Metric | Definition |
|---|---|
| *Reliability* | Continuity of correct service.[31] |
| *Reliability at time t* | Probability that the system or component survives until time *t*. |
| *Failure rate (λ)* | Frequency with which a system or component fails, expressed in failures per unit of time. |
| *Mean time between failures (MTBF)* | Predicted elapsed time between inherent failures of a system or component during operation.[32] |
| *Mean time to failure (MTTF)* | Predicted elapsed time to failure of a system or component. Average time between observed failures. |
| *Mean time to repair (MTTR)* | Predicted elapsed time required to repair a failed system or component. |
| *Availability or mission capable rate* | Readiness for correct service.[31] Proportion of time a system is in a functioning condition.[27] |
| *Availability at time t* | Probability that a system or component is able to function at time *t*.[33] |
| *Probability of survival* | Probability that a system or component does not fail in a time interval (0; *t*].[33] |

## 2.3 | Reliability analysis/evaluation

Sun et al[23] proposed the reliability analysis of a CPS based on a binary decision diagram (BDD) state-space model. The reliability analysis approach is demonstrated on a CPS case study of an aircraft fuel management system. First, a continuous-time Markov chain (CTMC) state model of the CPS components was designed in order to obtain the state probability vector of each component. Then, they developed a BDD for a CPS scenario with three phases. The CPS reliability is calculated according to the probabilities of the paths (successful states of the task) identified through the BDD. This approach combines two models, CTMC and BDD, to analyze the CPS's reliability, but it is demonstrated on a simplified scenario, which does not consider important elements such as the dependencies between the phases and the cyber states of the components.

Nannapaneni et al[24] addressed the topic of reliability evaluation of CPSs. They noted that it is difficult to achieve because of the problems derived from CPS components dependencies/interactions. The authors focused on assessing software reliability in terms of associated hardware and software inputs. The hardware failures are modeled using their failure rates, while the software failures are analyzed when the inputs are not within the expected ranges. The overall CPS failure probability is calculated as the union of the failure probabilities taking into account the CPS's operational and timing constraints. The proposed reliability analysis aims at supporting the design phase of the CPS, especially for component selection and runtime reconfiguration. The aforementioned authors[24] also proposed a model based on an optimization problem capable of reconfiguring the CPS in order for it to satisfy a maximum ratio of reliability to cost constraints. The proposed reliability analysis and reconfiguration model are demonstrated on a smart parking CPS.

Murtadha et al[25] presented a reliability model based on the Markov imbeddable structure (MIS) technique, which provides an overall CPS reliability analysis. In their analysis, the CPS's probability of failure is calculated based on the reliability of CPS's components and on the enumeration of CPS's binary states (1– functional, 0 – failed). The proposed reliability analysis is illustrated using a smart grid case study.

Zhang et al[26] addressed the challenge of assessing vulnerabilities, which can be found in the cyber networks of a power CPS. These vulnerabilities can be exploited by cyberattacks, which decrease the reliability and availability of information. The authors proposed and evaluated a Bayesian attack graph model of vulnerabilities to determine the probability that an intruder will successfully reach the target condition and a Bayesian attack graph model of communication links to estimate the probability of a successful intrusion on these links.

Wu et al[27] proposed a framework for benchmarking the reliability of a CPS. Failure Analysis and Reliability Estimation (FARE) provides an evaluation and an estimation of the overall CPS reliability using its historical data and a mechanism for the continuous monitoring and evaluation of the reliability for runtime enhancement. The framework was applied to a case study CPS for smart building management. Their experimental results show that the framework successfully estimates several reliability metrics based on the Weibull failure distribution and presents them in a graphical form.

Research studies on analysis or evaluation of CPS reliability are the most common in the current literature. Early approaches that address this topic can be found in the work of Bessani et al,[28] where a simulation framework for reliability analysis of a power CPS that incorporates the operator's response time during service restoration is presented. Another interesting approach is presented in the work of Li and Kang,[29] where a strategy for reliability testing and evaluation of CPSs is presented. Guo et al[30] presented the assessment of the reliability impact of monitoring functions within a power CPS.

Reliability evaluation uses a series of metrics (as shown in Table 1), which can also be applied to CPSs. These metrics can characterize the CPS's components or the entire CPS.

# 3 | REVIEW OF RELIABILITY TECHNIQUES PROPOSED FOR CPSs

The previous section reveals the importance of reliability in CPSs. In all CPS application domains, data reliability, reliability assurance, and reliability analysis need to be addressed, and in all these areas, the requirements concerning this aspect do not really differ. The development of new techniques for assuring CPSs' reliability combined with new metrics for its evaluation is of great importance.

## 3.1 | Techniques for data reliability

The data collected within a CPS is stored in databases, and through their analysis, useful and proactive information that ensures proper and reliable execution of the system can be provided to system operators. Data analysis is performed by using computational intelligence which combines machine learning, data mining, statistical and probabilistic analysis and other intelligent techniques.[9] These techniques handle missing or noisy data and find complex relations between different pieces of information.

The analysis of the data collected by the CPS should be achieved in real-time using dedicated components/subsystems and should follow a specific sequence of steps: (1) data preparation, which includes the processes related to data selection, cleaning, and construction; (2) data modeling, which involves the proper model selection and its design; and (3) data reliability evaluation. Several approaches for CPS data analysis were presented in other works,[8,9,34] in which resilient data aggregation algorithms, a combination of machine learning, data mining, statistical analysis and self-tuning techniques, and time series data mining algorithms were used to guarantee CPS data reliability.

**TABLE 2** Summary of the research efforts on recently proposed reliability techniques in CPSs

| Reliability Approach | Research Effort | Proposed Reliability Techniques | Description | Drawbacks, Issues, and Limitations | Results Achieved |
|---|---|---|---|---|---|
| *Data reliability* | Approaches to achieve resilience[8] | Data aggregation algorithms | Formalizes the notion of attack resilience through resilient sensor placement, network topology and data aggregation | Resource constraints of the devices and of the network have to be considered | New online data quality analysis based on computer intelligence (data aggregation and mining algorithms, machine learning techniques) |
| | Method for automated online evaluation of the CPS data[9] | Combination of machine learning, data mining, statistical analysis, and self-tuning techniques | Evaluates data quality of the system workflow Provides human operators feedback for improving reliability | There are some types of CPS that do not allow any kind of actions taken by human operators | |
| | Data mining model for CPS data prediction[34] | Time series data mining algorithms | Analyzes CPS historical data and predicts the missing information (such as lost data due to exhaustion of batteries, extreme environmental conditions, or communication protocol limitations) | The volume of data used for constructing the model has to be large to obtain an accurate forecasting | |
| *Reliability assurance* | Preliminary model for failure detection [11] | Combination of mailbox mechanism and sensor's data fusion | Takes into account system heterogeneity, the interaction between the CPS layers, real-time algorithms, and CPS's autonomous functions | Does not consider failure isolation, recovery, and prevention | New CPS components/ subsystems for failure and intrusion detection, isolation, recovery, and prevention at runtime |
| | Intrusion detection and response system for detecting and responding to different attacks at runtime [12] | Stochastic Petri Nets | Describes the behavior of the CPS in the presence of intrusions Maps a specification into a state machine and measures the deviation from good states in order to detect intrusion | Does not consider intrusion isolation | |
| | Real-time cascading failures prevention algorithm[22] | Adaptive multi-agent system algorithm | Uses self-adaptable techniques to increase the CPS reliability | Does not consider failure detection, isolation and recovery | |
| | Cascading failure prediction method[35] | Combination of multi-agent techniques and hybrid genetic algorithms | Investigates the behaviors of cascading failures to further study their prediction and defense | Does not consider failure detection, isolation, and recovery | |

**TABLE 2** (Continued)

| Reliability Approach | Research Effort | Proposed Reliability Techniques | Description | Drawbacks, Issues, and Limitations | Results Achieved |
|---|---|---|---|---|---|
| *Reliability analysis/ evaluation* | Assessment of CPS reliability[23] | Combination of two models (CTMC and BDD) | Takes into account the probabilities of successful states of the task, identified through the combination of CTMC and BDD | Does not consider the dependencies between the proposed CPS phases and the cyber states of the components | New analysis/ evaluation models used in the design phase, which take into account the interdependencies between all the CPS components (physical, cyber, network) New reliability metrics |
| | Assessing software reliability in terms of associated hardware and software inputs[24] | Optimization problem | Calculates the overall CPS failure probability taking into account the dependencies between CPS components Offers help for component selection and runtime | Presents reliability analysis solutions only for the CPS design phase Does not propose new reliability metrics | |
| | Reliability model for overall CPS′s reliability analysis[25] | Markov analysis (MIS) | Calculates the CPS Calculates the CPS probability of failure based on the system components′ reliability and on the enumeration of its binary states | Does not propose new reliability metrics | |
| | Assessment of vulnerabilities found in cyber networks of a CPS[26] | Bayesian graphs | Addresses the issue of cyber-attacks within CPSs Determines the probability that an intruder will successfully reach the target condition Estimates the probability of successful intrusion on the communication links | Does not consider the failures of the CPS hardware and software components | |
| | Framework for benchmarking the reliability of CPSs[27] | Simulation frameworks | Uses historical data to evaluate and estimate the overall CPS reliability | Uses only Weilbull failure distribution to estimate several reliability metrics Does not propose new reliability metrics | |

## 3.2 | Techniques for reliability assurance

Fault detection and isolation (FDI) techniques represent the means through which system reliability is assured. These techniques are divided into two categories, namely model-based and signal processing. The first category uses a mathematical- or knowledge-based system model, while the second category uses high-performance data analysis to extract relevant information about the fault. FDI mechanisms are not sufficient in the context of CPSs where new mechanisms for detection, isolation, recovery, and prevention at runtime have to be implemented to enable them to deal with different threats (faults, errors, failures) and external attacks. Also, the implementation of techniques such as self-adaptation, self-organizing, and fault-tolerance capabilities increase CPSs' reliability.

The combination of the two FDI categories leads to specific CPS solutions for failure and intrusion detection as shown by the approaches described in the works of Guo et al[11] and Mitchell and Chen.[12] Multi-agent systems can also be used to detect and prevent failures while providing support for the self-adaptation and self-organization of CPS. To prevent the occurrence of cascading failures, prediction methods based on adaptive, multi-agent system algorithms,[22] and multi-agent and hybrid genetic algorithms[35] have also been proposed.

## 3.3 | Techniques for the analysis/evaluation of reliability

The evaluation of the reliability level of an engineering system is an important aspect that must be considered early on during the design phase. Reliability evaluation requires system modeling in terms of its components' failures. The reliability block diagram (RBD) and the fault tree analysis (FTA) are combinatorial methods that describe the structure of the system's reliability characteristic as a framework made up of independent components. Usually, system components in a reliable structure are dependent, and thus, their modeling is performed using bivariate and multivariate lifetime distributions such as the BVE distribution of Marshall and Olkin,[36] which is suitable for modeling common-cause failures, or Freund's model,[37] which can be applied when a failed component increases the other component's failure rate.[38] Other commonly used reliability analysis techniques are Markov models and Stochastic Petri Nets (SPNs). These state-based techniques enable the explicit modeling of complex relationships between system components and the evaluation of complex repair and maintenance strategies.[39] Markov models specify the dependence of failure or repair characteristics of individual components on the state of the system, while a continuous-time Markov chain (CTMC) is the reference model for reliability assessment. SPNs are used to check the existence of unsafe states of the system and calculate the reliability metrics for assumptions of transient or steady system states.[39]

All these techniques can be applied to assess the reliability feature within CPSs. However, specific characteristics of CPSs require new evaluation models, usually starting with the design phase, which take into account the dependencies between all the CPS components (physical, cyber, network) as well as the definition of new reliability metrics. Several early attempts were presented in other works,[23-30] in which new models based on Markov analysis, Bayesian graphs, and on the optimization problem, and simulation frameworks were used to evaluate CPSs' reliability.

The analysis of the previously presented techniques for assuring reliability in CPSs shows that a combination of those can represent a solution, but at the same time, it does not meet all the requirements imposed by this type of systems (Table 2).

## 4 | PROPOSED APPROACH WITH AGENT TECHNOLOGY

The literature review highlights several solutions for the three approaches for achieving enhanced levels of CPS reliability. As Table 2 shows, all these previously proposed approaches have strengths and weaknesses. In this section, we present our proposed solution, ReliaCPS, which addresses the reliability assurance challenges in CPSs. In contrast to past solutions, ReliaCPS covers not only the prevention of failures but also failure detection and isolation. We achieve these goals based on an in-depth analysis of the behavior of the failures to better predict them in the future and of the defensive actions against their occurrence. The analysis is performed in the design/modeling phase of the CPS and acts as a knowledge base for several software components, which make up our proposed CPS subsystem for failure detection and prevention. The subsystem operates in parallel with the CPS operations and also uses its historical data to take recommended control actions which can be changed or overridden by human operators. The historical data is stored in a centralized database rather than a distributed one because it will cost too much to maintain replication and real-time update between nodes. The multi-agent paradigm is used to implement the CPS subsystem because it allows the division of tasks and resources in order for them to be managed by different proactive, autonomous software entities, called agents, which use the rules stored in a knowledge base to achieve their goals. Currently, this approach leverages the benefits of high performance computers. ReliaCPS provides more flexibility to users because it is a high-level approach that could be used by them without requiring a strong knowledge of networking or operating system (OS) programming. ReliaCPS introduces two components: the CPS multi-agent subsystem for failure detection and prevention and the knowledge base (KB), which determines the behavior of the agents.

We have identified four types of agents in our CPS subsystem architecture, which we describe below. To develop an efficient solution, we restrict the number of different agent types, and we limit the communication among them. Therefore, the workflow of the multi-agent system assumes that, first, the agents analyze the possibility of cooperation by sending and receiving messages. If the agents are not performing any tasks, then they start their cooperation; otherwise, they continue to execute the activities that they are already doing. The cooperation among the agents assumes the completion of their tasks. If those cannot be successfully completed, they attempt to start a new common activity. More details about the agents' types, their tasks, and the cooperation between them are provided below.

### 4.1 | Proposed multi-agent subsystem architecture

Our proposed CPS multi-agent subsystem uses the following agents:

- A *Data Processing Agent for Diagnosis* (DPAD) that interprets the CPS historical data.
- A *Diagnosis Agent* (DA) that detects and localizes the failures using the knowledge base.
- A *Data Processing Agent for Prevention* (DPAP) that processes the CPS historical database.
- A *Prevention Agent* (PA) that performs proactive actions using the knowledge base.

The DPAD and the DPAP periodically analyze the data collected by the sensors, which is then stored into a database. The DPAD collaborates with the DA and the DPAP with the PA. DA and PA access the knowledge database, retrieving the information required for detecting and localizing the failures and for taking preventive actions. Subsequently, they update the CPS's historical database with information concerning the detected failure, the probability of failure, and the decision that must be taken.

*Cooperation between the DPAD and the DA.*

The DPAD processes the data in the CPS historical database for validation purposes and provides information about its accuracy, timeliness, and consistency. When an unexpected value is detected, the DPAD sends a request to the DA so that it investigates the type of failure that produced the exception, and if the DA is not busy with another diagnosis task, the request is accepted. In this case, the DPAD provides additional information on

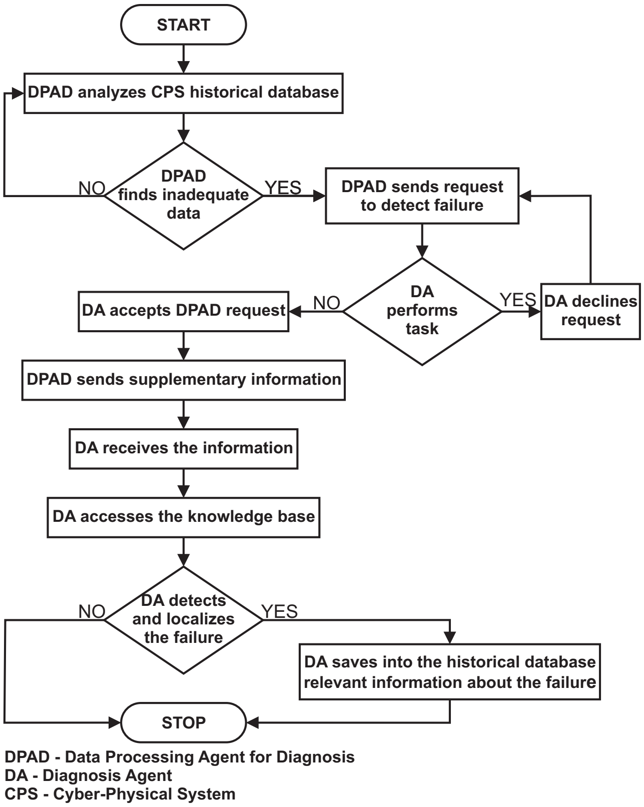

START
DPAD analyzes CPS historical database
DPAD finds inadequate data
NO
YES
DPAD sends request to detect failure
DA performs task
NO
YES
DA declines request
DA accepts DPAD request
DPAD sends supplementary information
DA receives the information
DA accesses the knowledge base
DA detects and localizes the failure
NO
YES
DA saves into the historical database relevant information about the failure
STOP
DPAD - Data Processing Agent for Diagnosis
DA - Diagnosis Agent
CPS - Cyber-Physical System

**FIGURE 2** Workflow of the DPAD and the DA agents within the CPS multi-agent subsystem

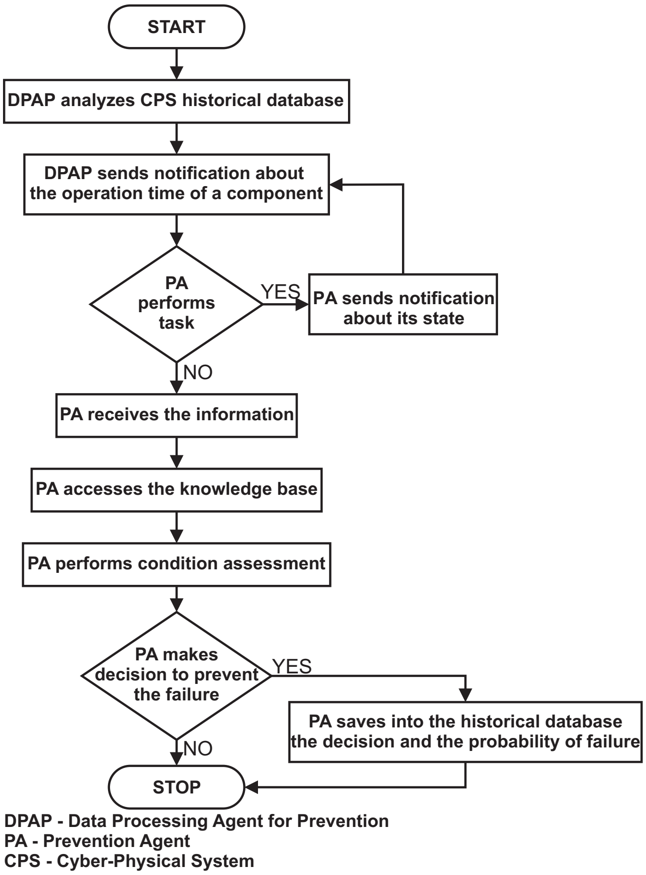

START
DPAP analyzes CPS historical database
DPAP sends notification about the operation time of a component
PA performs task
YES
PA sends notification about its state
NO
PA receives the information
PA accesses the knowledge base
PA performs condition assessment
PA makes decision to prevent the failure
YES
PA saves into the historical database the decision and the probability of failure
NO
STOP
DPAP - Data Processing Agent for Prevention
PA - Prevention Agent
CPS - Cyber-Physical System

**FIGURE 3** Workflow of the DPAP and the PA agents within the CPS multi-agent subsystem

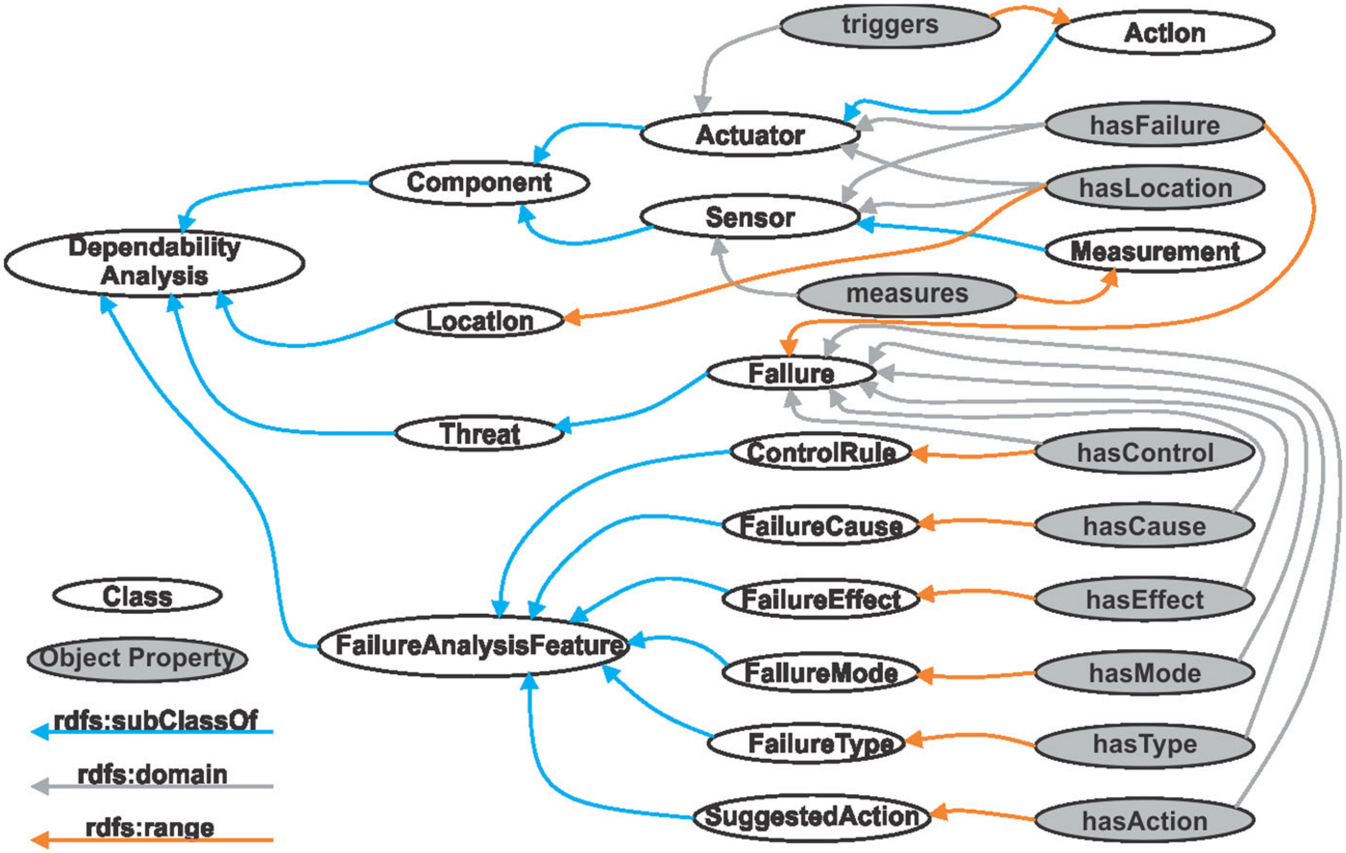


**FIGURE 4** Representation in the form of an ontology of the KB used by the CPS multi-agent subsystem

the failure. If the DA is busy, a reject is received from the DA, and a new request is sent at a later time. When a failure is detected and localized by the DA, the related information consisting of its name, type, detection time, the damaged CPS component, and the countermeasures are saved into the CPS historical database so that other specific non-agent software components within the system have access to it and can inform the human operators about the exception that occurred. Figure 2 shows the cooperation between the DPAD and the DA.

*Cooperation between the DPAP and the PA.*

The DPAP periodically queries the CPS historical database and notifies the PA of the operation time of the hardware components in the system since the last maintenance activities. The PA in turn analyzes the received data and performs condition assessment, estimating the probabilities for CPS hardware components' failures, and, if necessary, it schedules maintenance and replacement actions such as component repairing, component replacement, or schedule an inspection time. The new information provided by the PA is then saved into the CPS historical database, and the human operators are informed about the required actions. Figure 3 shows the cooperation between the DPAP and the PA.

### 4.2 | Knowledge base

The knowledge base (KB) used by the DA and the PA contains information about the CPS hardware components (eg, name, date of last revision, period between revisions, location, measurements, triggered actions, additional information), their possible failures (eg, type, mode, cause, effect), the control actions that are already implemented, and the suggested actions that have to be carried out to mitigate failures immediately. All this information has to be obtained based on a careful and detailed analysis that enables the proper detection and localization of failures in a short period of time and the fast selection of appropriate preventive actions.[40] The knowledge base is represented as an ontology that can be inherently used by the multi-agent subsystem. The implementation and validation of the ontology is the subject of previous work and are described in the work of Sanislav and Mois.[41] Figure 4 shows the ontology classes, the relations between them, and their object properties.

## 5 | EVALUATION OF THE PROPOSED APPROACH

To demonstrate the usability and the performance of the proposed CPS multi-agent subsystem architecture, we integrated it within a case study CPS for monitoring indoor ambient parameters such as temperature (T), relative humidity (RH), pressure (P), light intensity and carbon dioxide ($CO_2$). It is a well-known fact that cities consume a large percentage from the world's energy[42] and that they have a significant contribution to the carbon impact over the environment. It is therefore expected that cyber-physical systems will be developed to address this issue in the future. One such system is the one taking care of the microclimate inside large office buildings,[43,44] where the ambient parameters are monitored and properly managed for assuring the comfort of the occupants while minimizing the energy consumed.

### 5.1 | CPS case study

Figure 5 presents an overview of the proposed CPS case study. The physical layer of this system is composed of embedded devices with sensor capabilities spread over the floors and rooms of an office building.[45] These sensors can collect temperature-related data, relative humidity levels, illumination intensities, absolute pressures, and carbon dioxide concentration levels. In order to make the deployment and operation as easy as possible, the embedded devices are powered by batteries but can have power harvesting capabilities.[46] The collected data is sent via the User Datagram Protocol (UDP) to the CPS controller, where a software application, namely EnvDataAcquisition, interprets them and saves the interpretation results into an SQLite database. EnvDataAcquisition is a software application running on a central device with a high processing power and high connectivity capacity and performs the monitoring functions in this CPS case study and provides the relevant information to the CPS operators/end-users via Hypertext Transfer Protocol (HTTP). Network connections use the IEEE 802.11 set of standards because this is almost ubiquitous in smart buildings and provides the communications infrastructure. The CPS human operators act as actuators in the proposed scenario because they have the ability to execute control actions according to the information delivered by the CPS. All software applications are executed using the CPS controller hardware, the central host, represented in this case by a high-performance computer.

A simple analysis of a CPS case study consisting of ten sensor nodes and one central host in terms of reliability demonstrates several failures. A critical failure occurs when the battery of one of the embedded devices gets depleted. It is worth noting that the system does not have any redundant equipment and the CPS does not store any information regarding the battery status of its physical components equipped with sensor capabilities. Another system failure that can occur is data loss during transmissions. The table and the chart presented in Figure 6 show the daily failure rates of a CPS embedded device, which collects T, RH, $CO_2$, P, and light intensity every 10 minutes, for one week. The failure rate is associated in this case with the lost data packets due to the UDP communication protocol. On the first day of operation, four records were not saved into the database, on the second day of operation, five records were left out, and so on.

It is worth noting that the CPS does not detect or prevent failures.

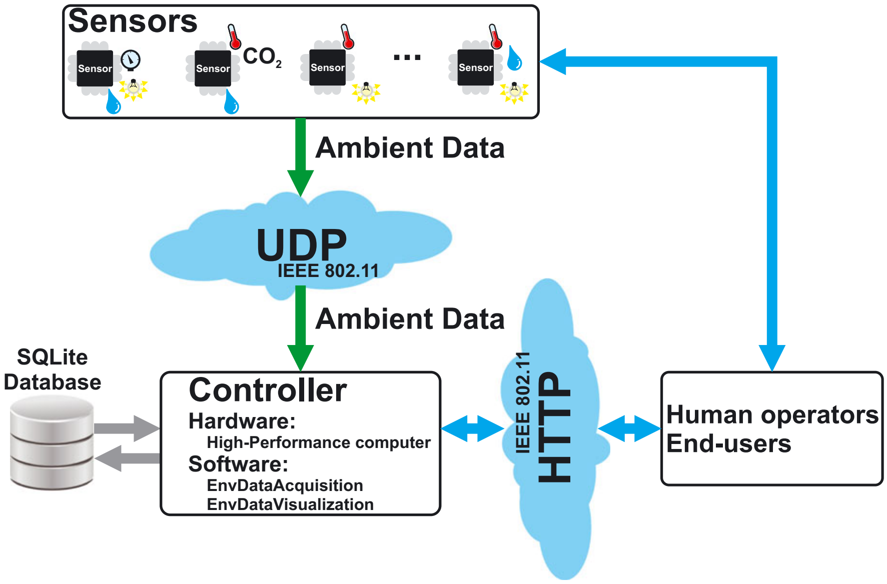


**FIGURE 5** Architecture overview of the CPS case study for monitoring indoor ambient parameters

| Day | Failure rate |
|---|---|
| 1 | 4 |
| 2 | 5 |
| 3 | 4 |
| 4 | 4 |
| 5 | 5 |
| 6 | 3 |
| 7 | 5 |

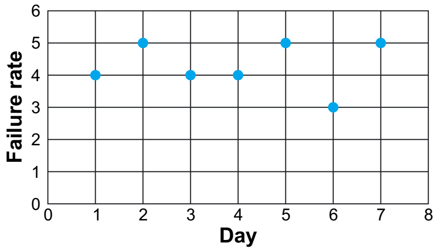


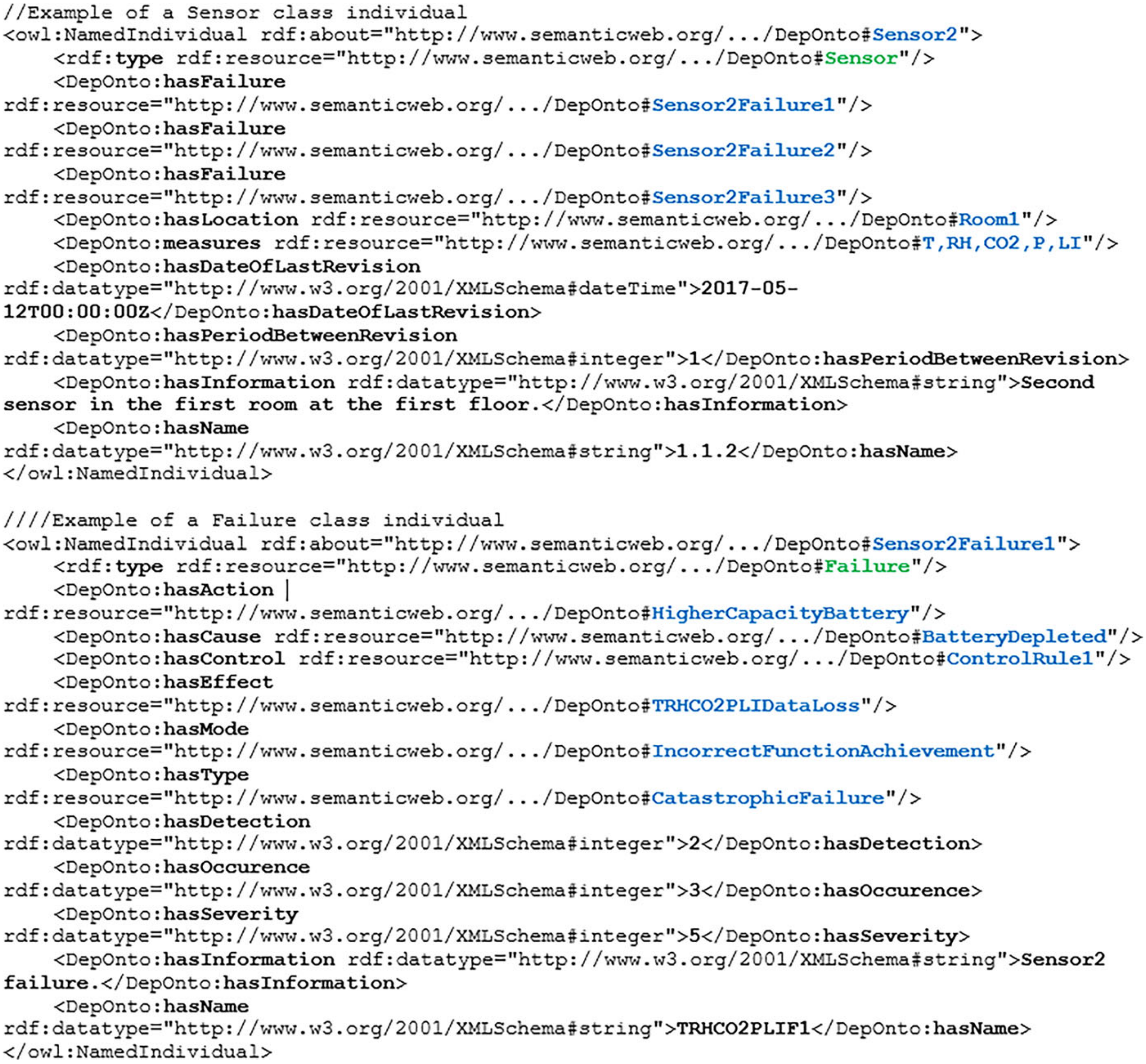
**FIGURE 6** Daily failure rate of an embedded device in the CPS case study

## 5.2 | CPS case study subsystem for reliability assurance

First of all, we performed an analysis of the failures of the CPS's hardware components. The results of the analysis were transferred to a knowledge base represented as an ontology developed in Protégé.[47] Figure 7 shows a part of the DepOnto ontology that contains two individuals: a CPS embedded device description, called Sensor 2, and one of its failures, namely Sensor2Failure1. The text in green defines the class to which the entity/event belongs. The text in black is used for the object and data types' properties of each entity, and the blue color for naming an entity belonging to another class. This last one is related to a main entity through an object type's property. More details are provided in the work of Sanislav and Mois.[41]

Second, a historical database (DB) of the CPS case study is implemented in SQLite, which initially only stored into its tables the collected ambient data. It has been modified to store information regarding the system's failures (name, type, detection time, damaged CPS component, actions to be performed) and maintenance aspects (date of last revision, period between revisions, probability of failure, actions performed) of its components.

The proposed multi-agent CPS subsystem uses the historical data that was stored in the DB and the KB to perform data analysis and recommend control actions that can be changed or overridden by human operators. We describe their tasks in the context of the CPS case study next. DPAD filters and validates the data stored in DB and processes them in order to inform the DA about its accuracy, timeliness, and consistency. Based on this information and with the support of the KB, the DA detects and localizes the failure. When the DA detects a failure, it saves the information related to the failure in the DB, from where the EnvDataVisualization application can retrieve it to inform the operators about the problem. DPAP processes the DB information to inform the PA about the operation times of the CPS components. PA helps the CPS human operators to prevent



```
//Example of a Sensor class individual
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/.../DepOnto#Sensor2">
    <rdf:type rdf:resource="http://www.semanticweb.org/.../DepOnto#Sensor"/>
    <DepOnto:hasFailure
rdf:resource="http://www.semanticweb.org/.../DepOnto#Sensor2Failure1"/>
    <DepOnto:hasFailure
rdf:resource="http://www.semanticweb.org/.../DepOnto#Sensor2Failure2"/>
    <DepOnto:hasFailure
rdf:resource="http://www.semanticweb.org/.../DepOnto#Sensor2Failure3"/>
    <DepOnto:hasLocation rdf:resource="http://www.semanticweb.org/.../DepOnto#Room1"/>
    <DepOnto:measures rdf:resource="http://www.semanticweb.org/.../DepOnto#T,RH,CO2,P,LI"/>
    <DepOnto:hasDateOfLastRevision
rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2017-05-
12T00:00:00Z</DepOnto:hasDateOfLastRevision>
    <DepOnto:hasPeriodBetweenRevision
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</DepOnto:hasPeriodBetweenRevision>
    <DepOnto:hasInformation rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Second
sensor in the first room at the first floor.</DepOnto:hasInformation>
    <DepOnto:hasName
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">1.1.2</DepOnto:hasName>
</owl:NamedIndividual>

////Example of a Failure class individual
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/.../DepOnto#Sensor2Failure1">
    <rdf:type rdf:resource="http://www.semanticweb.org/.../DepOnto#Failure"/>
    <DepOnto:hasAction |
rdf:resource="http://www.semanticweb.org/.../DepOnto#HigherCapacityBattery"/>
    <DepOnto:hasCause rdf:resource="http://www.semanticweb.org/.../DepOnto#BatteryDepleted"/>
    <DepOnto:hasControl rdf:resource="http://www.semanticweb.org/.../DepOnto#ControlRule1"/>
    <DepOnto:hasEffect
rdf:resource="http://www.semanticweb.org/.../DepOnto#TRHCO2PLIDataLoss"/>
    <DepOnto:hasMode
rdf:resource="http://www.semanticweb.org/.../DepOnto#IncorrectFunctionAchievement"/>
    <DepOnto:hasType
rdf:resource="http://www.semanticweb.org/.../DepOnto#CatastrophicFailure"/>
    <DepOnto:hasDetection
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">2</DepOnto:hasDetection>
    <DepOnto:hasOccurence
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">3</DepOnto:hasOccurence>
    <DepOnto:hasSeverity
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">5</DepOnto:hasSeverity>
    <DepOnto:hasInformation rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Sensor2
failure.</DepOnto:hasInformation>
    <DepOnto:hasName
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">TRHCO2PLIF1</DepOnto:hasName>
</owl:NamedIndividual>
```

**FIGURE 7** Part of the DepOnto ontology which exemplifies the description of one embedded device in the CPS case study and of one of its failures

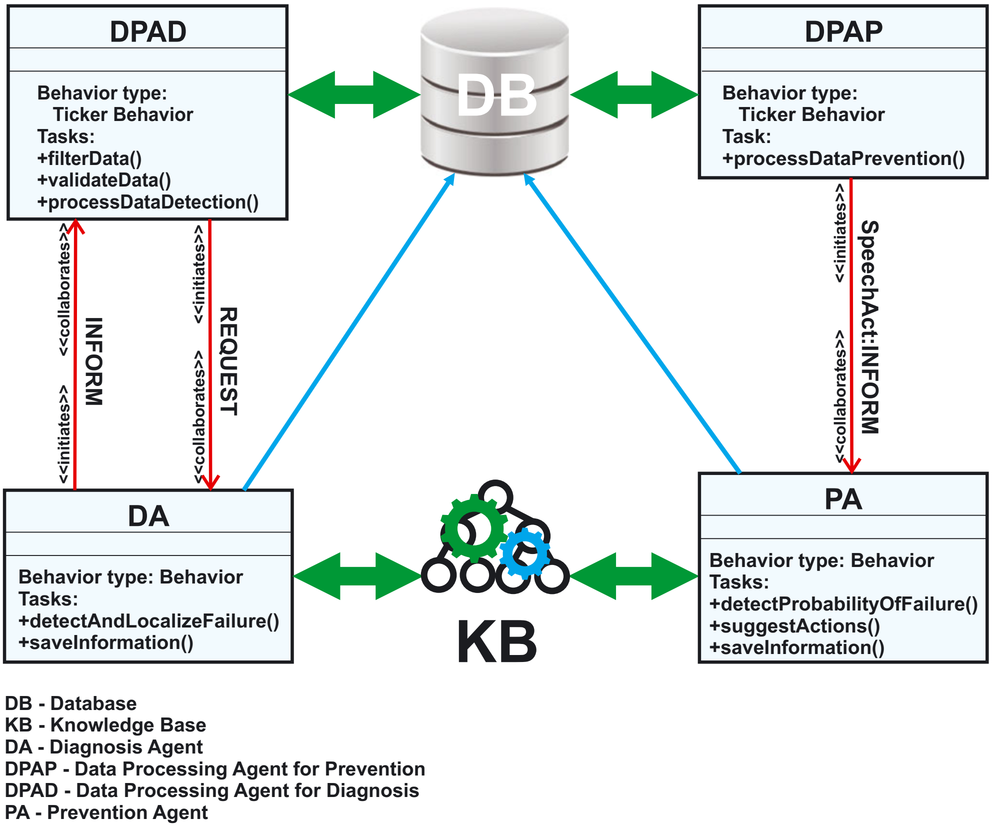


**FIGURE 8** Overview of the CPS multi-agent subsystem classes and their communication methods

failures by performing the following proactive actions: condition assessment, determination of the probability of failure, decision making (eg, schedule repair/replace, schedule next inspection). It saves the new information related to the probability of failure of each component and the possible actions that have to be taken in order to prevent the failure in the DB. Figure 8 shows the components of the CPS multi-agent subsystem along with their main tasks and interactions.

The agents' behaviors are implemented by the EnvAgents application using Java Agent DEvelopment Framework (JADE),[48] which simplifies the implementation of multi-agent systems in conformity with FIPA specifications. The application is hosted by the central host.

## 5.3 | Experimental evaluation procedure

To evaluate the proposed approach, we consider a simplified CPS case study consisting of ten sensor nodes ($S_1, \ldots, S_{10}$), which collect T, RH, $CO_2$, P and light intensity, and one central host. The sensor nodes are the CPS units, and the minimum number of units required to have a successful operating state is 10 because the operation of all the nodes is mandatory to have a CPS that achieves its goal. The acquisition rates taken into consideration for the ambient parameters are 10 minutes for $S_1$, $S_2$, $S_3$, $S_4$, $S_5$ and 30 minutes for $S_6$, $S_7$, $S_8$, $S_9$, $S_{10}$, respectively. These values are suitable for the monitoring of low-rate varying ambient parameters. The type of failure analyzed (detection, localization and prevention) through the current experimental procedure is battery depletion. The values corresponding to the following CPS reliability metrics: reliability at time $t$, $\lambda$, MTBF, MTTF, and MTTR (see Table 1) are calculated before (initial test) and after (final test) the use of the multi-agent subsystem. The unit failure rate ($\lambda$) is calculated as the number of failures per million hours (FPMH). The formulas used to calculate the reliability at time $t$, effective MTBF, and MTTF, for a mission time $T = 24$ hours, are as follows:

$$R(t) = e^{-\lambda t} \tag{1}$$

$$\text{MTBF} = \frac{\int_0^T R(t)\,\mathrm{d}t}{1 - R(T)} \tag{2}$$

$$\text{MTTF} = \int_0^\infty R(t)\,\mathrm{d}t. \tag{3}$$

### 5.3.1 | Initial test – Without the proposed solution

Previous experiments have shown that the battery lifetime for the two acquisition rates considered are 4200 and 12 600 hours, respectively.[45] Table 3 presents the reliability metrics values of the CPS under consideration before the inclusion of the multi-agent subsystem and of the DepOnto ontology. The system reliability is zero (approximately) at 5803 hours. MTTR is estimated at 1 hour for each unit if the failure is caused by battery depletion.

### 5.3.2 | Final test – With the proposed solution

The CPS multi-agent subsystem has proven its efficiency when the acquisition rate of sensor nodes is 30 minutes. The DPAP continuously processes the historical data in order to determine the operation time of each CPS component. The PA uses this information to calculate the probability of failure and, in this case, to approximate the time when the battery is depleted then recommends its replacement and saves this information in the database. The CPS human operators are informed, and if they act accordingly, the replacement of the battery can be performed between two data acquisitions. By knowing the lifetime of all the hardware components that make up the sensor devices within the CPS, the PA can prevent failure occurrences that can be caused by those, in a way similar to the one presented above. The immediate result of this action is the improvement of the $\lambda$ and MTTR metrics. Additionally, the CPS multi-agent subsystem improves reliability in cases when the acquisition rate of sensor nodes is 10 minutes. The DPAD processes the data, and when if it finds out that data is missing, it informs the DA. The DA detects and localizes the possible failure and saves the information into the database. EnvDataVisualization displays the detected failure and the actions suggested to mitigate it to the CPS human operators, who can replace the battery in a time interval shorter than the one given by MTTR (˜ 30 minutes). Table 4 presents the values of the reliability metrics of the CPS considered after the inclusion of the multi-agent system and of the DepOnto ontology. The system reliability is zero at (approximately) 7704 hours, and the CPS is restored to “as good as new” every 24 hours.

The other considered type of failure can be caused by data losses during transmission as we have discussed in Section 4. These losses can be detected by the DA when it is informed by the DPAD that there is one missing piece of information in the data time series. The failure to store the data

**TABLE 3** Values of reliability metrics of the CPS case study calculated before the use of the proposed solution

| CPS Unit | Unit Failure Rate, FPMH | Unit MTBF, h | Unit Reliability at 24 Hours | Unit MTTR, h | R(24)* | Effective MTBF, h | MTTF, h |
|---|---|---|---|---|---|---|---|
| $S_1$ | 238.095 | 4200 | 0.9943 | 1 | 0.9626 | 630 | 630 |
| $S_2$ | 238.095 | 4200 | 0.9943 | 1 | | | |
| $S_3$ | 238.095 | 4200 | 0.9943 | 1 | | | |
| $S_4$ | 238.095 | 4200 | 0.9943 | 1 | | | |
| $S_5$ | 238.095 | 4200 | 0.9943 | 1 | | | |
| $S_6$ | 79.36 | 12 601 | 0.9981 | 1 | | | |
| $S_7$ | 79.36 | 12 601 | 0.9981 | 1 | | | |
| $S_8$ | 79.36 | 12 601 | 0.9981 | 1 | | | |
| $S_9$ | 79.36 | 12 601 | 0.9981 | 1 | | | |
| $S_{10}$ | 79.36 | 12 601 | 0.9981 | 1 | | | |

*Overall probability of successful system operation for 10 units, where a minimum of 10 are required = the result of units' reliability multiplication.

**TABLE 4** Values of reliability metrics of the CPS case study calculated after using the proposed solution

| CPS Unit | Unit Failure Rate, FPMH | Unit MTBF, h | Unit Reliability at 24 hours | Unit MTTR, h | R(24)* | Effective MTBF, h | MTTF, h |
|---|---|---|---|---|---|---|---|
| $S_1$ | 238.095 | 4200 | 0.9943 | 0.50 | 0.9717 | 836 | 836 |
| $S_2$ | 238.095 | 4200 | 0.9943 | 0.50 | | | |
| $S_3$ | 238.095 | 4200 | 0.9943 | 0.50 | | | |
| $S_4$ | 238.095 | 4200 | 0.9943 | 0.50 | | | |
| $S_5$ | 238.095 | 4200 | 0.9943 | 0.50 | | | |
| $S_6$ | 1 | 1 000 000 | 1 | 0.50 | | | |
| $S_7$ | 1 | 1 000 000 | 1 | 0.50 | | | |
| $S_8$ | 1 | 1 000 000 | 1 | 0.50 | | | |
| $S_9$ | 1 | 1 000 000 | 1 | 0.50 | | | |
| $S_{10}$ | 1 | 1 000 000 | 1 | 0.50 | | | |

*Overall probability of successful system operation for 10 units, where a minimum of 10 are required = the result of units' reliability multiplication.

**TABLE 5** Comparison of the CPS reliability metrics' values obtained before and after the use of the proposed solution

| Reliability Metrics | Initial Test | Final Test | Improvement |
|---|---|---|---|
| Effective MTBF, h | 630 | 836 | 32.69% |
| MTTF, h | 630 | 836 | 32.69% |
| MTTR, h | 1 | 0.5 | 50% |
| *R(t)*, T = 24 Hours | 0.9626 | 0.9717 | 0.94% |

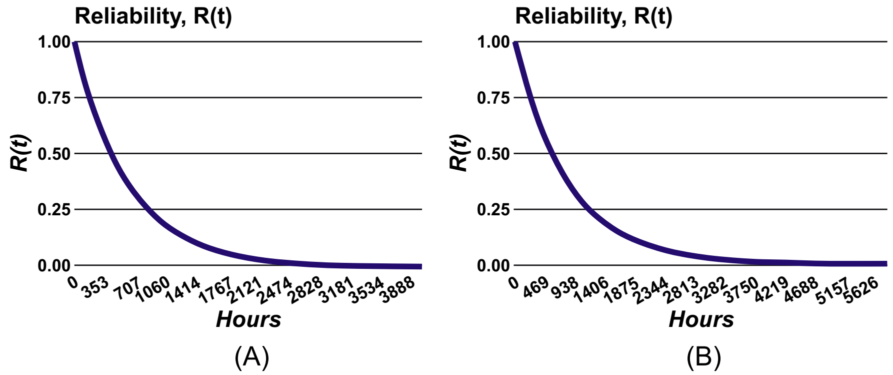


**FIGURE 9** Values of the reliability function (R(t)) for the CPS case study. A, Initial test; B, Final test

in the database cannot be prevented because it depends on the communication protocol (in our case, UDP), which is unreliable due to the absence of acknowledgement packets. However, there are several solutions to mitigate this type of failure. One solution is the use of series-parallel redundant structures of sensors in each CPS node. Another possible solution would be the implementation of computational intelligence techniques capable of predicting the data time series.

## 6 | PERFORMANCE EVALUATION RESULTS

Table 5 shows that the previous experiments demonstrate an improvement of the reliability metrics considered.

Figure 9 presents the reliability function over time, calculated using the MTBF value, and a time period of interest of 24 hours as inputs in both tests. The time period in which the reliability value reaches zero is longer when the CPS multi-agent subsystem is used. The CPS is also restored to the “as good as new” state every 24 hours.

When the CPS comprises a large number of sensor nodes with each node having a measurement rate of 30 minutes, the value of R(t) rises, reaching the maximum value of 0.9998 for a mission time of 24 hours. When the measurement rate of the nodes in the CPS is set to 10 minutes, MTTR is significantly improved.

## 7 | CONCLUSION

In this work, we have reviewed recently proposed CPS solutions for data reliability, reliability assurance, and assessment used in different domains, and we analyzed recently proposed techniques through which they have been achieved. The agent technology solution shows promise, especially in reliability assurance, because it can solve problems through cooperation and tasks distribution and can enable taking decisions in the context of incomplete information. Therefore, we proposed a multi-agent–based approach to increase the reliability of CPSs. Our multi-agent–based architecture implements mechanisms for failure detection and prevention by using the collected data and an ontology where all the CPS failures are described. We evaluated the proposed solution using a CPS case study for ambient monitoring, which takes into consideration several reliability metrics ($\lambda$, reliability function, effective MTBF, MTTF, MTTR). The results obtained demonstrate improvements for all the metrics we have considered. However, we are aware that further experiments should be performed to verify the behavior of the proposed solution for different types of failures identified during the system analysis phase, which we plan to do as part of our future work.

### ACKNOWLEDGMENT

This research was supported by a grant from the Technical University of Cluj-Napoca through the research Contract Number 1998/12.07.2017, Internal Competition CICDI-2017. We would like to thank the editors and the anonymous reviewers for their valuable comments, which helped us to improve the quality and presentation of this paper.

## ORCID

*Teodora Sanislav* http://orcid.org/0000-0002-3019-004X
*Hacène Fouchal* http://orcid.org/0000-0002-9584-3566

## REFERENCES

1. Wolf W. Cyber-physical systems. *Computer*. 2009;42(3):88-89.
2. Poovendran R. Cyber–physical systems: close encounters between two parallel worlds [Point of View]. *Proc IEEE*. 2010;98(8):1363-1366.
3. Zanero S. Cyber-physical systems. *Computer*. 2017;50(4):14-16.
4. Zeadally S, Jabeur N, eds. *Cyber-Physical System Design With Sensor Networking Technologies*. London, UK: IET Press; 2016.
5. CPSE Labs. Cyber-Physical Systems. http://www.cpse-labs.eu/cps.php. Accessed September 1, 2017.
6. Avizienis A, Laprie J-C, Randell B. Fundamental Concepts of Dependability. [Technical Report]. Los Angeles, CA: University of California; 2001.
7. Barnum S, Sastry S, Stankovic JA. Roundtable: reliability of embedded and cyber-physical systems. *IEEE Secur Priv*. 2010;8(5):27-32.
8. Abbas W, Laszka A, Koutsoukos X. Resilient wireless sensor networks for cyber-physical systems. In: Zeadally S, Jabeur N, eds. *Cyber-Physical System Design With Sensor Networking Technologies*. London, UK: IET Press; 2016:239-267.
9. Wu LL. Improving System Reliability for Cyber-Physical Systems. [Technical Report]. New York, NY: Columbia University; 2015. https://doi.org/10.7916/D8JQ10GW. Accessed September 1, 2017.
10. Babaei M, Ghassemieh H, Jalili M. Cascading failure tolerance of modular small-world networks. *IEEE Trans Circuits Syst II Exp Briefs*. 2011;58(8):527-531.
11. Guo A, Yu D, Du H, Hu Y, Yin Z, Li H. Cyber-physical failure detection system: survey and implementation. Paper presented at: 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems; 2016; Chengdu, China.
12. Mitchell R, Chen IR. Effect of intrusion detection and response on reliability of cyber physical systems. *IEEE Trans Rel*. 2013;62(1):199-210.
13. Huang Z, Wang C, Nayak A, Stojmenovic I. Small cluster in cyber physical systems: network topology, interdependence and cascading failures. *IEEE Trans Parallel Distrib Syst*. 2015;26(8):2340-2351.
14. Zhang Z, An W, Shao F. Cascading failures on reliability in cyber-physical system. *IEEE Trans Rel*. 2016;65(4):1745-1754.
15. Zhu Y, Yan J, Sun Y(L), He H. Revealing cascading failure vulnerability in power grids using risk-graph. *IEEE Trans Parallel Distrib Syst*. 2014;25(12):3274-3284.
16. Huang Z, Wang C, Stojmenovic M, Nayak A. Characterization of cascading failures in interdependent cyber-physical systems. *IEEE Trans Comput*. 2015;64(8):2158-2168.
17. Zhang X, Zhan C, Tse CK. Modeling the dynamics of cascading failures in power systems. *IEEE Trans Emerg Sel Topics Circuits Syst*. 2017;7(2):192-204.
18. Chattopadhyay S, Dai H, Eun DY, Hosseinalipour S. Designing optimal interlink patterns to maximize robustness of interdependent networks against cascading failures. *IEEE Trans Commun*. 2017;65(9):3847-3862.
19. Cai Y, Cao Y, Li Y, Huang T, Zhou B. Cascading failure analysis considering interaction between power grids and communication networks. *IEEE Trans Smart Grid*. 2016;7(1):530-538.
20. Rahnamay-Naeini M, Hayat MM. Cascading failures in interdependent infrastructures: an interdependent Markov-chain approach. *IEEE Trans Smart Grid*. 2016;7(4):1997-2006.
21. Zhang X, Liu D, Zhan C, Tse CK. Effects of cyber coupling on cascading failures in power systems. *IEEE Trans Emerg Sel Topics Circuits Syst*. 2017;7(2):228-238.
22. Babalola AA, Belkacemi R, Zarrabian S. Real-time cascading failures prevention for multiple contingencies in smart grids through a multi-agent system. *IEEE Trans Smart Grid*. 2018;9(1):373-385.
23. Sun X, Huang N, Wang B, Zhou J. Reliability of cyber physical systems assessment of the aircraft fuel management system. Paper presented at: IEEE 4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems; 2016; Wanchai, Hong Kong.
24. Nannapaneni S, Mahadevan S, Pradhan S, Dubey A. Towards reliability-based decision making in cyber-physical systems. Paper presented at: IEEE International Conference on Smart Computing; 2016; St Louis, MO.
25. Albasrawi MN, Jarus N, Joshi KA, Sarvestani SS. Analysis of reliability and resilience for smart grids. Paper presented at: IEEE 38th Annual Computer Software and Applications Conference; 2014; Vasteras, Sweden.
26. Zhang Y, Wang L, Xiang Y, Ten CW. Power system reliability evaluation with SCADA cybersecurity considerations. *IEEE Trans Smart Grid*. 2015;6(4):1707-1721.
27. Wu L, Kaiser G. FARE: A framework for benchmarking reliability of cyber-physical systems. Paper presented at: 2013 IEEE Long Island Systems, Applications and Technology Conference; 2013; Farmingdale, NY.
28. Bessani M, Fanucchi RZ, Delbem ACC, Maciel CD. Impact of operators' performance in the reliability of cyber-physical power distribution systems. *IET Gener Transm Distrib*. 2016;10(11):2640-2646.
29. Li Z, Kang R. Strategy for reliability testing and evaluation of cyber physical systems. Paper presented at: 2015 IEEE International Conference on Industrial Engineering and Engineering Management; 2015; Singapore.
30. Guo J, Wang Y, Guo C, Dong S, Wen B. Cyber-physical power system (CPPS) reliability assessment considering cyberattacks against monitoring functions. Paper presented at: IEEE Power and Energy Society General Meeting; 2016; Boston, MA.
31. Avizienis A, Laprie JC, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans Dependable Secure Comput*. 2004;1(1):11-33.
32. Jones JV. *Integrated Logistics Support Handbook*. 3th ed. New York, NY: Sole Press; 2006.
33. Rausand M, Høyland A. *System Reliability Theory: Models, Statistical Methods, and Applications*. 2nd ed. Hoboken, NJ: John Wiley & Sons; 2004.

34. Sanislav T, Merza K, Mois G, Miclea L. Cyber-physical system dependability enhancement through data mining. Paper presented at: IEEE International Conference on Automation, Quality and Testing, Robotics; 2016; Cluj-Napoca, Romania.

35. Li J. A cascading failure prediction method in power system based on multi-agent and hybrid genetic algorithm. *Appl Mech Mater*. 2014;494-495:1598-1601.

36. Marshall AW, Olkin I. A multivariate exponential distribution. *J Am Stat Assoc*. 1967;62(317):30-44.

37. Freund JE, Olkin I. A bivariate extension of the exponential distribution. *J Am Stat Assoc*. 1961;56(296):971-977.

38. Wang R-T. Reliability evaluation techniques. In: Mathew J, Shafik RA, Pradhan DK, eds. *Energy-Efficient Fault-Tolerant Systems*. New York, NY: Springer; 2014:11-97.

39. Bernardi S, Merseguer J, Petriu D. Dependability analysis techniques. *Model-Driven Dependability Assessment of Software Systems*. Berlin, Germany: Springer; 2013:73-90.

40. Sanislav T, Zeadally S, Mois G, Fouchal H. Multi-agent architecture for reliable cyber-physical systems (CPS). Paper presented at: IEEE Symposium on Computers and Communications; 2017; Heraklion, Greece.

41. Sanislav T, Mois G. A dependability analysis model in the context of cyber-physical systems. Paper presented at: IEEE 18th International Carpathian Control Conference (ICCC); 2017; Sinaia, Romania.

42. Khatoun R, Zeadally S. Smart cities: basic concepts, architectural issues, and research opportunities. *Commun ACM*. 2016;59(8):46-57.

43. Foster TW, Bhatt DV, Hancke GP, Silva B. A web-based o-ce climate control system using wireless sensors. *IEEE Sensors J*. 2016;16(15):6104-6113.

44. Javed A, Larijani H, Ahmadinia A, Emmanuel R, Mannion M, Gibson D. Design and implementation of a cloud enabled random neural network-based decentralized smart controller with intelligent sensor nodes for HVAC. *IEEE Internet Things J*. 2017;4(2):393-403.

45. Mois G, Sanislav T, Folea SC. A cyber-physical system for environmental monitoring. *IEEE Trans Instrum Meas*. 2016;65(6):1463-1471.

46. Shaikh F, Zeadally S. Energy harvesting in wireless sensor networks: a comprehensive review. *Renew Sustain Energy Rev*. 2016;55:1041-1054.

47. Protégé Documentation. https://protege.stanford.edu/support.php. Accessed September 10, 2017.

48. Bellifemine FL, Caire G, Greenwood D. *Developing Multi-Agent Systems with JADE*. 1st ed. Hoboken, NJ: John Wiley & Sons; 2007.

**How to cite this article:** Sanislav T, Zeadally S, Mois GD, Fouchal H. Reliability, failure detection and prevention in cyber-physical systems (CPSs) with agents. *Concurrency Computat Pract Exper*. 2018;e4481. https://doi.org/10.1002/cpe.4481