A Generic Framework For Capturing Reliability in Cyber-Physical Systems

Nazakat Ali, Manzoor Hussain, Youngjae Kim and Jang-Eui Hong
Department of Computer Science, Chungbuk National University, Republic of South Korea
jehong@chungbuk.ac.kr

ABSTRACT

Research Purpose: Cyber-Physical Systems solve complex problems through their tight integration between the physical and computational components. Therefore, the reliability of a complex system is the most critical requirement for the cyber-physical system because an unreliable system often leads to service disruption, property damage, financial losses and sometimes lead to fatality. In order to develop more reliable CPS, this paper proposes a generic framework for reliability modeling and analysis for our ongoing work on cyber-physical systems. **Research Method:** This paper, at first defines an architecture for general CPS which is comprised of three layers; environment layer, communication layer, and computational layer. Secondly, we formalize a reliability model for the architectural components, and then propose a framework for the reliability of CPS with the consideration of how to capture the reliability. **Results:** Based on the research method, we demonstrate the proposed framework with an illustrative example by using different reliability values from “offshore and onshore reliability data” library. We confirmed that the reliability model covers almost all possible reliabilities required to general cyber-physical systems.

CCS Concepts

- **Cyber-Physical Systems** → **Reliability Architecture engines**
- **Framework** → **Generic framework.**

Keywords

Cyber-Physical Systems; System Architecture; Reliability; System Modeling.

1. INTRODUCTION

Cyber-Physical Systems (CPSs) have emerged as a new engineering technique which is used to solve complex problems in various fields of information technology such as smart buildings, smart grids, self-driving vehicles, smart manufacturing systems, smart transportation, oil and gas industries, nuclear power plants and avionics [1]. CPSs are composed of physical components such as sensors and actuators, communication network, and cyber components such as computation and control, and monitoring unit. The cyber and physical components are highly intertwined to achieve a specific task. Reliability of CPSs has become a critical concern for researchers because a single failure or a malfunction in the system may lead to property damage, financial loss or even loss of human lives. Therefore, CPSs are not designed

to work in a controlled environment, they must be designed to operate in a robust manner to handle unpredicted conditions and adaptable to system failures [1].

In CPSs, sensors are utilized to collect environmental or system data and transmit this data through a network to central computation and control unit for further processing [2]. The information can be extracted from the processed data and then change in the environment can be responded through actuators.

In this paper, we propose an architecture for a general CPS which includes an environmental (physical) layer, communication and network layer, and computation and control layer (cyber). The environmental layer has sensor module and actuator module. The cyber layer includes computation and control module and reliability model. The computation and control module is composed of hardware and software components. The software and hardware have a strong interaction with each other in order to complete a task. The reliability model is proposed to measure the CPS reliability. The reliability model has sensor reliability, actuator reliability, communication reliability, data reliability, and computation and control unit reliability which includes hardware reliability, software reliability, and software-hardware interaction reliability.

In summary, we make following contributions:

- First, we propose a generic framework for capturing reliability in CPS includes data reliability.
- Second, we demonstrate the proposed framework with an illustrative example by using different reliability values from a “offshore and onshore reliability data” library [3].
- Software-hardware interaction is critical in computing CPS reliability. Therefore, we also present a survey on how software fails due to hardware failure (software-hardware interaction), and how hardware (hardware-software interaction) fails due to interaction with a faulty software.

The rest of paper is organized as follow: Section 2 discusses the reliability modeling for CPS. Section 3 presents an illustrative example to explain our proposed architecture. In Section 4, we investigate related work and Section 5 concludes this study.

2. MODELING CPS WITH RELIABILITY

We propose CPS architecture in order to model its reliability characteristics. CPS is consisting of cyber space and physical (environment) space. Physical space and cyber space communicate through a communication network. In our CPS architecture, the CPS is proposed to have reliability model as well because proper functioning of complex systems like CPS is vital due to fact that a single failure in CPS may lead to catastrophic consequences such as financial loss, and probably human death.

This work is licensed under a Creative Commons Attribution International 4.0 License.

ESSE 2020, November 6–8, 2020, Rome, Italy
© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7762-1/20/11...\$15.00

<https://doi.org/10.1145/3393822.3432331>

The proposed architecture as shown in Fig. 1 has three layers: 1) physical layer, which includes the sensor module and actuator module, 2) communication network layer, where the communication medium between physical and cyber space is defined. The medium can be a wireless network, a wired network or a hybrid communication network, 3) computation and control or cyber layer which includes computation and control module and a reliability model. The computation and control module consists of software and hardware components. The hardware and software have a strong interaction with each other in order to complete a task. By analyzing modules of the target CPS, we have defined a reliability model which should be included in CPS.

The sensor module of our CPS architecture may consist of a group of sensors that monitor the physical (environment) world and capture the data and transmit it to the cyber world. The actuator module of our CPS architecture may consist of a group of actuators to reflect the response for change in the environment. The data captured through sensors is transmitted to cyber world (computation and communication module) through communication network. The network performs two-way communication between cyber and physical world such as sensors and actuators using Machine-to-Machine (M2M) communication. The M2M is considered to be highly reliable and low latency link for data transmission [4]. The computation and control module of cyber layer is composed of software and hardware components. The software component may include data processing component to process the information received from sensors and sends that status information to the control algorithms component of software to generate the required control signal for actuators. The monitoring component maintains historical data of all the components in order to monitor the health of associated components. The actuators respond to the environment upon receiving the required control signals.

In order to measure the reliability of a CPS, we propose a reliability model for the CPS. The reliability model is consisting of sensors' reliability, to ensure the behavior of sensors, actuators' reliability, to ensure the behaviour of actuators, communication reliability, to ensure the network is working fine, data reliability to ensure the

consistency and completeness of input data from sensors. The computation and control module consists of hardware reliability, software reliability, and the software-hardware interaction reliability.

2.1 Reliability Modeling and Analysis for CPS

For safety-critical systems, continuous availability is an essential requirement and software reliability is one of important components of continuous system availability. Reliability is defined as the probability that software or hardware will perform its intended task for a specific time under certain conditions [5]. Reliability is related with time to failure, or its reciprocal, and failure rate. The reliability of whole CPS is determined by computing the reliability of each component. In CPS, its components are designed to work independently. However, in order to achieve a common goal, the components have to work collectively and correctly. Consequently, the failure of a single component may lead to failure of whole system. Therefore, the actual CPS has a series of configurations in terms of its reliability. The reliability of a system component is computed by the probability of successful functioning $P(X_i)$ of that component independently. The reliability of entire system R_{sys} can be the intersection of the probability $P(X_i)$ of each system component, represented as equation (1).

$$R_{sys} = P(X_1 \cap X_2 \cap X_3 \cap \dots \cap X_n) \\ = P(X_1)P(X_2|X_1)P(X_3|X_1X_2) \dots P(X_n|X_1X_2 \dots X_{n-1}) \quad (1)$$

Where R_{sys} is system reliability and $P(X_i)$ is the probability that component i works correctly.

When the failure of a system component affects the failure rates of rest of the components (i.e., the life distribution characteristics of the other components change when one component fails), the conditional probabilities in the equation (1) must be taken into account. However, in the case of independent components, equation (1) can be re-written as follows;

$$R_{sys} = P(X_1)P(X_2)P(X_3) \dots P(X_n) \quad (2)$$

Figure 1. A generic architecture for reliability modeling and analysis for CPS.

Given that the failure rate for each system component is independent and constant of their usage of time, equation (2) can be written as equation (3).

$$R_{sys} = \prod_{i=1}^n R_i \quad (3)$$

Where R_i is the reliability of each independent component i and n is the number of components per operation in the system.

The redundancy is the common technique to improve the reliability of a system [6]. Adding redundant components in the system may lead to high reliability. The redundant component is used when the same working component was failed. In this case, the reliability of the system with redundant components is given in equation (4).

$$R_{sys} = 1 - \prod_{i=1}^n (1 - R_i) \quad (4)$$

Where n is the number of redundant components doing the same job.

The reliability of a single component is determined in terms of failure rate as a function of time as mentioned in equation (5).

$$R = e^{-\lambda t} \quad (5)$$

Where R is the reliability of the component and λ is the failure rate of the component at time t . When we consider the failure rate λ as constant mean value (i.e., exponentially distributed time t between failures), the reliabilities as mentioned in equation (3) and (4) are determined on the basis of exponential distribution as shown in equation (5). The mean time between failure (MTBF) of the component can be expressed as equation (6).

$$MTBF = \frac{1}{\lambda} \quad (6)$$

2.2 Reliability Modeling for Individual Modules

In this section, we focus on the reliability modeling for individual modules of target CPS. Our CPS architecture has sensor modules, actuator modules, communication modules, and computation and control module.

1) Sensor modules

Let's suppose that a system requires additional sensors for a single task. Let n be the total number of sensors, mn be the minimum number of sensors required to perform the task. The possible ways to select mn from n is expressed by $nCmn$. Subsequently, the reliability for any combination can be formulated according to the selection.

Assume that a CPS has a temperature sensor, a humidity sensor, a fire sensor, and a rain sensor to facilitate the inhabitants of smart home [7]. Sensitivity, sampling, noise, frequency, accuracy, and failure rate are considered to parameters to determine sensor performance. However, the failure rate determines the reliability of a sensor as mentioned in equation (7).

$$R_{sensor} = e^{-\lambda_s t} \quad (7)$$

2) Actuator module

The widespread metrics to measure the performance of actuators are speed, durability, torque, force, and operation conditions. The reliability of actuators can be determined by considering the above

parameters. Hence, the reliability for actuators can be expressed as equation (8).

$$R_{actuators} = e^{-\lambda_a t} \quad (8)$$

Where, λ_a is corresponding failure rate of an actuator.

3) Communication network reliability

The communication network is the backbone of CPSs which acts as an information carrier. It carries information from physical layer to cyber layer and back to physical layer. There are a number of communication techniques to carry information such as wireless communication, wired communication, and hybrid communication (combination of wired and wireless communication). However, M2M type of communication is widely used in CPSs. M2M architecture has mainly three domain such as M2M device domain, network domain, and application domain [8]. Data from physical layer goes through various levels [2] and converted into information which is used by business processing engines for decision making. M2M gateway is considered communication network node which is used for interfacing among group of networks. This gateway uses various protocols, translators, impedance matching devices, signal isolators and fault isolators to ensure reliability.

4) Computation and control module

As shown in Fig. 1, the computation and control module is consisting of software and hardware. To measure the reliability of software and hardware, Weibull distribution and Markov process are usually used [9]. The software part of computation and control is independent of time while the hardware module is depending on time. In CPS, the software and hardware are highly intertwined. Therefore, the failure can be divided into three categories such as 1) hardware failure, 2) software failure and 3) software-hardware interaction failure.

The hardware failures can be better handled with Weibull distribution because it is robust to determining distribution depending on parameters such as scale, shape, and location. More specifically, the Weibull is more convenient and reliable for real-time applications. Therefore, the reliability for hardware component is expressed as in equation (9).

$$R_{Hardware}(t) = e^{-\lambda t^\beta} \quad (9)$$

Where, λ is combined failure rate for each hardware component, β is the shape parameter of the Weibull distribution model. The hardware component in computation and control unit refers to storage and computation devices e.g. CPU etc.

The software reliability growth model (SRGM) is an important model to measure software reliability. The most general SRGM model is the Non-Homogeneous Poisson Process (NHPP) SRGM with mean value function [10]. The base model for SRGM is derived from G-O model [11] and modifies only hypothesis of G-O model. The SRGM based NHPP model is highly recommended for controlling, predicting, and analyzing the reliability of a software. Therefore, the accumulative number of failures at the time t is expressed as:

$$P\{N(t) = n\} = \frac{\{m(t)\}^n}{n!} \exp\{-m(t)\}, n = 0, 1, \dots, t \geq 0.$$

Where $m(t)$ is the mean value function in a time t . The failure intensity function tells the failure occurrence rate at time t . The failure intensity function expressed below shows relationship between failure intensity function and mean value function.

The failure rate for the software system can be expressed as follows:

$$\lambda(t) = \frac{d}{dt} m(t) = ab \exp[-bt]$$

Therefore, reliability for software component can be written as equation (10).

$$R_{software}(t) = e^{-(m(t+x)-m(t))} = e^{-\lambda_{st}} \quad (10)$$

Software interacts with hardware to perform a task [12]. Therefore, software-hardware interaction sometimes leads to software failure. The software-hardware failures occur due to the manipulations in the hardware configurations. It is also possible that the manipulations in software configurations may lead to hardware failure while interacting with hardware. This scenario is called software-hardware interaction failure. We also considered this software-hardware interaction in reliability modeling as shown in Fig.1. Weibull distribution model [9] is proposed to compute the reliability of software-hardware interaction. Hence,

$$\lambda_{SH}(t) = \alpha_{SH} * \beta_{SH} * t^{\beta_{SH}-1} \quad \alpha, \beta > 0, t \geq 0$$

Where λ_{SH} is the rate of software-hardware (SH) interaction failure, α_{SH} and β_{SH} are scale and shape parameters. We see that the failure rate is proportional to power $\beta_{SH} - 1$ of time t .

The mean value function of software-hardware interaction failure will be scale parameter times power shape parameter of time t as shown below:

$$m(t) = \alpha_{SH} * t^{\beta_{SH}}$$

Therefore, the reliability for interaction failure can be expressed as:

$$R_{SH}(t) = e^{-(m(t+x)-m(t))} = e^{-(\alpha_{SH} * (t+x)^{\beta_{SH}} - \alpha_{SH} * t^{\beta_{SH}})} \quad (11)$$

Hence, reliability for computation and control module would be formulated in equation (12).

$$R_{CC}(t) = R_{software}(t) + R_{Hardware}(t) + R_{SH}(t) \quad (12)$$

Therefore, reliability estimation for the entire CPS without data reliability can be modeled as equation (13).

$$R_{CPS} = R_{CC} * R_{Actuator} * R_{sensors} * R_{CN} \quad (13)$$

5) Data reliability

Data reliability means the quality characteristics of the data that must be processed by the software system. Data processed in CPS applications can be categorized with (1) internal data processed by the application itself, and (2) data transmitted from other external systems. Their representative examples include data input from various types of sensors, and messages from interactive external systems. All of these data must be taken into account in common way whether data quality or data reliability is appropriate for processing in CPS system. Namely, data reliability is an evaluation of whether the data meets its intended use and can be expressed in terms of accuracy, consistency, and completeness [13].

Especially, data reliability in CPS applications also includes timeliness in addition to these basic quality characteristics. This is a factor that must be available when data is required. These characteristics are shown in Table 1 and can be expressed as shown in equation (14).

$$R_{Data} = D_c + D_a + D_i + D_t \quad (14)$$

Table 1. Critical factors of data reliability on CPS applications.

Data Quality	Description	Measurement Approach
Completeness (Dc)	Is there any data gap?	Count the omitted data items based on input specification
Accuracy (Da)	Are all the data within the acceptable range? Do data fit into the usual profile?	Count the data items which overrun the range and precision of the specified values
Consistency (Di)	Are all the data in the same format?	All data item to be processed should be matched its types and formats with the specified data types and formats
Timeliness (Dt)	Do data arrive on time?	Count the data items which violate the specified time limits.

Therefore, the reliability estimation for the entire CPS with data reliability can be modeled as the equation (15).

$$R_{CPS} = R_{CC} * R_{Actuator} * R_{sensors} * R_{CN} * R_{Data} \quad (15)$$

3. RELIABILITY COMPUTATION WITH AN ILLUSTRATIVE EXAMPLE

We assume that a CPS is designed with off the shelf components available in the market. Each CPS component with different implementations and configurations can be selected while using off the shelf components. The components with the highest reliability can be chosen after reliability analysis. We used a dataset from [3] and calculated the reliabilities for each component as shown in Fig.2. The proposed CPS has five sensors (S1, S2, S3, S4, S5) and three actuators (A1, A2, A3) with additional redundant five sensors and three actuators to improve reliability. The proposed system also has two communication networks and two computation and control units instead of one to ensure the reliability. Each module (sensor module, actuator module, communication, and network module, computation and control module) has replicated components. Therefore, equation (4) was used to calculate reliability for parallel redundant components. The reliability for entire CPS, without considering data reliability, was then calculated using equation (13) by selecting highest reliable component from each module. Hence, entire CPS reliability was 0.9538 without considering data reliability.

Figure 2. Component selection for highest reliability.

4. RELATED WORK

This section provides related work about the reliability modeling of CPS, and then presents a survey for interaction (software-hardware

and hardware-software interaction) reliability. Table 2 shows the summary of related work with their focus and approach overview.

4.1 Reliability Modeling for CPS

For safety-critical systems, continuous availability is a firm requirement and software reliability is one of important components of continuous system availability. A number of researchers considered reliability for CPSs. Leon Wu et al. [5] presented a failure analysis and reliability estimation framework to benchmark reliability for CPSs. The proposed framework provides methods and metrics for reliability analysis, reliability estimation and operational availability to benchmark CPS reliability. This framework provides a general but accurate overview for representing CPS reliability. The authors introduced three additional reliability metrics to cover specific evaluation scenarios.

Nannapaneni et al. [14] proposed a framework in order to formulate CPS reliability evaluation as a dependency issue derived from functional requirements, software components, and physical system dependencies. The proposed framework is argued to analyze CPS reliability in the early design phase.

Koc et al. [15] proposed a study that models the reliability for CPSs. The authors model reliability for software, hardware, sensors, and actuators. However, the proposed architecture is a simple one that does not cover all kinds of reliabilities such as data reliability.

Some researchers have proposed reliability benchmark [16] for specific CPSs, i.e. wireless sensor networks and wind plant. Wu et al. [17] have presented a hierarchical model for reliability analysis with the focus on the integrated modular avionics system. However, it is domain specific framework which cannot be applied for developing general CPSs considering application domains.

4.2 Interaction Reliability

The equation (13) is one of the general models to compute reliability for entire cyber-physical systems. However, different researchers have categorized the software-hardware interaction failures in two ways [12]: 1) Hardware-driven software interaction failures, which are software failures caused by hardware, 2) software-driven hardware failure is the failure caused by faulty software. In this section we survey the literature that considered the above two cases for computing software-hardware interaction reliability.

Hardware-driven interaction failures are caused due to a fault in hardware or changes in hardware configuration. Teng et al. [18] have argued that software systems fail due to hardware component fails, or software component fails or hardware-software interaction failure. The authors have used Weibull distribution model for hardware component and NHPP for software components to estimate reliability. The failure distribution of hardware-software interaction was modeled using Markov chain. The authors computed system reliability by taking product of independent software failure, independent hardware failure, and hardware-software interaction failure probability. However, the Markov Improving System Reliability for Cyber-Physical Systems chain model has a number of limitations such as some critical information e.g. the number of cumulative detected failures, test time, and software release time cannot be used during system testing.

Sinha Roy et al. [19] presented a reliability framework to estimate reliability for Phasor Measurement Units. The authors used Weibull distribution model for hardware component failure, NHPP to model the software components failure, and Markov chain to model the

hardware-software interaction failures. However, the authors used Monte Carlo simulation to validate their proposed model.

5. CONCLUSION

Cyber-Physical systems have grown significantly in various application domains. The CPSs systems are widely believed as safety critical systems. For safety-critical systems, continuous availability is a necessity and software reliability is one of important components of continuous system availability. In this work in progress, we present an architecture to capture reliability for CPSs. The proposed architecture has the reliability characteristics for sensor, actuator, communication and network, data, software and hardware, and interaction (software-hardware, hardware-software). Using the reliability models presented, we are working to measure the reliability for Automatic Incident Detection System (AIDS), as one of CPS.

In our future work, we are planning to determine reliability for entire CPS using equation (15) as defined in our proposed framework. Also, we are considering to developing safety model in quantitative manner.

6. ACKNOWLEDGMENT

This research was supported by the NRF of Korea funded by the MSIT (NRF-2017M3C4A7066479).

7. REFERENCES

- [1] Lee, E. A. Lee, 2008. Cyber Physical Systems: Design Challenges. *11th Int. Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, Orlando, FL, 2008, pp. 363-369.
- [2] Wu L.L. 2015. Improving system reliability for cyber-physical systems (Doctoral dissertation, Columbia University. pp.3-6, 2015.
- [3] Sintef. N. T. N. U, 2015. Offshore and onshore Reliability Data." OREDA Handbook, 6th ed, DNV, Oslo , 2015: Available: <https://www.oreda.com>
- [4] Mehmood, Y., Haider, N., Imran, M., Timm-Giel, A., & Guizani, M. 2017. M2M communications in 5G: state-of-the-art architecture, recent advances, and research challenges. *IEEE Communications Magazine*, 55(9), 194-201.
- [5] Wu. L and Kaiser.G, 2013. FARE: A framework for benchmarking reliability of cyber-physical systems. *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, 2013, pp. 1-6.
- [6] Flehinger. B. J, 1958 Reliability Improvement through Redundancy at Various System Levels. in *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 148-158, Apr. 1958.
- [7] Ali .N, and Hong.J, 2018. Failure Detection and Prevention for Cyber-Physical Systems Using Ontology-Based Knowledge Base. *Computers* 7.4 (2018): 68. doi: 10.3390/computers7040068
- [8] Jiafu. W, Chen. M, Xia. F, Li. D, Zhou. K. 2013. From machine-to-machine communications towards cyber-physical systems. *Comput. Sci. Inf. Syst*, 2013, pp. 1105-1128.
- [9] Alessandro. B, 2017. Reliability & Availability of Repairable Systems. *Reliability Engineering*. Springer, Berlin, Heidelberg, 2017. Pp. 169-310.
- [10] Wu. C, Zhu.X and Liu. J, 2008. The SRGM Framework of Integrated Fault Detection Process and Correction Process.

- International Conference on Computer Science and Software Engineering, Hubei, 2008, pp. 679-682.
- [11] Goel. A. L and Okumoto.K, 1997. Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures. in IEEE Transactions on Reliability, vol. R-28, no. 3, pp. 206-211, Aug. 1979.
 - [12] Feng. E, Zheng. J and Liu. C, 2014. An integrated reliability model of hardware-software system. 10th International Conference on Reliability, Maintainability and Safety (ICRMS), Guangzhou, 2014, pp. 577-580.
 - [13] Wang, R. Y., & Strong, D. M. 1996. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4), 5-33
 - [14] Nannapaneni, S., Mahadevan, S., Pradhan, S., & Dubey, A. (2016, May). Towards reliability-based decision making in cyber-physical systems. In 2016 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1-6.
 - [15] Koc, H., Shaik, S. S., & Madupu, P. P. 2019. Reliability modeling and analysis for cyber physical systems. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0448-0451.
 - [16] Sun. X, Huang.N, Wang.B and Zhou. J 2014. Reliability of cyber physical systems assessment of the aircraft fuel management system. The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent, Hong Kong, 2014, pp. 424-428.
 - [17] Wu. Z, Huang.N, Zheng.X and Li.X , 2014. Cyber-physical avionics systems and its reliability evaluation. The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent, Hong Kong, 2014, pp. 429-433.
 - [18] Teng. X, Pham. H and Jeske. D. R, 2006. Reliability Modeling of Hardware and Software Interactions, and Its Applications. in IEEE Transactions on Reliability, vol. 55, no. 4, pp. 571-577, Dec. 2006.
 - [19] Roy. D. S, Murthy. C and Mohanta. D. K, 2015. Reliability analysis of phasor measurement unit incorporating hardware and software interaction failures. in IET Generation, Transmission & Distribution, vol. 9, no. 2, pp. 164-171, 29 1 2015.