# PhD Forum Abstract: A Robust Discrete Event Method for the Design of Cyber-Physical Systems*

Joseph Boi-Ukeme†
Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
joseph.boiukeme@carleton.ca

## ABSTRACT

The technological advancement in Cyber-Physical Systems (CPS) has evolved into sophisticated hardware, leading to systems that are complex and interconnected. This trend has made modern CPS susceptible to faults. We present a discrete event method developed using the Discrete Event System Specification (DEVS) to detect, diagnose, and accommodate CPS faults in Real-time. This includes Fault Detection and Diagnosis (FDD), Fault Tolerance (FT) in the control system, and Sensor Fusion in the Sensor system called SAFE

## CCS CONCEPTS

• Computing Methodologies • Modeling and Simulation

## KEYWORDS

Cyber-Physical Systems, FDD, Fault Tolerance

**ACM Reference format:**

Joseph Boi-Ukeme, 2020. PhD Forum Abstract: A Robust Discrete Event Method for the Design of Cyber-Physical Systems. In *Proceedings of The 18th ACM International Conference on Embedded Networked Sensor Systems (SenSys), Nov 16-19, 2020, Virtual Event, Japan.*

## 1 Introduction

The advancement in sensing, actuation, and control systems technology has led to the development of complex systems. These systems are characterized by the tight interconnectivity between hardware and software. Generally, such systems are fragile and prone to faults and failure. [1].

The theory of CPS was developed to ensure that there is better coordination between hardware and software. The design of cyber-physical systems involves three major disciplines: (1) control, (2) systems, and (3) software engineering. Each of these disciplines has its design principles and well-established methods for design and implementation. However, it is difficult to adopt any of these disciplines as a universal approach for the design of cyber-physical systems because the methods are domain-specific and not generic enough for CPS.

Modeling and Simulation (M&S) is a universal approach that gaining popularity because of its low development cost, short development time, as well as its ability to provide a framework for reusability and testing [2]. M&S also cut across the three disciplines and can connect the different aspects of cyber-physical systems [3]. However, current modeling and simulation techniques are unable to reproduce the same conditions in the environment that are very critical to the correct operation of a CPS system. The inability to explicitly model the environmental conditions increases the probability of the occurrence of faults, hence failures are bound to occur. Also, most modeling techniques do not explicitly describe how CPS faults and failures should be handled. [4]

To address the above-mentioned issues, we present a modeling technique that considers the environment as well as anticipates the occurrence of faults/failures and is equipped with the relevant tools to deal with these faults promptly.

## 2 A Method for FDD and FT in CPS

The proposed method is event-based, and it only triggers the fault tolerance algorithm when the values that fall out of range within the CPS are confirmed to be faults. Note that not all values that fall out of range are faults. For example, it could just noise or uncertainty. Within the CPS, we would define algorithms for fault tolerance. These algorithms would only be executed when a fault has been confirmed by the FDD scheme presented in this section. The scheme, shown in Figure 1, describes the method.

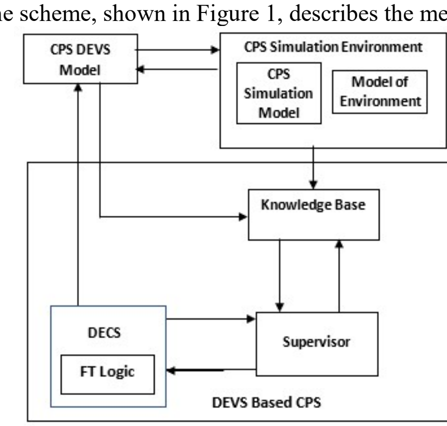


**Figure 1. FDD/FT Scheme for CPS Design**

The first component is the ***CPS DEVS Model***. It is a model of the CPS using DEVS, which is used for testing and to populate the Knowledge Base with known faults. Once this model has been specified and implemented, we can execute it in a simulation environment: the ***CPS Simulation Environment*** block. This is a model of the environment of the CPS that allows us to simulate the CPS DEVS Model. With these simulations, we can study the model's behavior in various scenarios. The CPS DEVS Model, which was already evaluated using the CPS Simulation Environment, is deployed into the target platform and it is transformed into the Discrete Event Control Software (DECS).

The ***DECS*** component allows us to perform further testing and calibration. We may need to go back and redefine the CPS DEVS Model as the testing of the DECS in the real world or the CPS Simulation Environment may reveal some potential Any fault information observed during testing and calibration of the DECS is stored in the Knowledge Base component.

The ***Knowledge Base (KB)*** is a database that holds information about the known set of faults that can occur in the CPS. The knowledge base evolves through the lifetime of the CPS because as new faults become known, information about these faults would be updated in the knowledge base.

The ***Supervisor*** is an intermediate component between the DEVS Based CPS and the KB that manages the fault detection process. When the DECS notifies values out of range, it confirms with the KB if a fault has occurred. If a fault occurred, it notifies the DECS to take the recommended actions about that fault determined by the ***FT Logic.***

## 3 A method for FT in Sensor Systems

Figure. 2 shows the sensor fusion framework developed to improve the reliability of the sensor system.

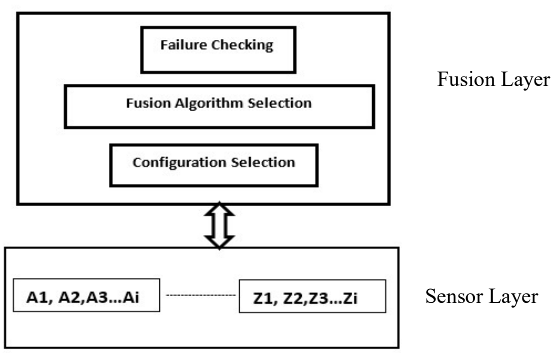


**Figure 2. SAFE Framework for FT in Sensor Systems**

The framework is organized into two layers. The first one is a ***Sensor Layer*** with various groups of sensors. In this layer, all the sensors for fusion are separated into groups. Sensors measuring the same property are grouped. The second one is the ***Fusion Layer***, which includes three blocks, namely: *Configuration Selection*, *Fusion Algorithm Selection*, and *Failure Checking*. The ***Configuration Selection*** block oversees the input decision for the *Fusion Algorithm Selection* block. The ***Fusion Algorithm Selection*** block implements streamlined algorithms that support that configuration from the previous block**.** The ***Failure Checking*** block checks the outputs of the *Sensor Fusion* layer to verify the correctness of the sensor information.

## 4 Case Study: Simulation Results

To test the FDD/FT scheme we implemented an example with a DECS with four models and one state variable each. Table 1. shows results obtained from a case study of the FDD/FD scheme with seven fault codes.

**Table 1. Knowledge Base with Detected Faults**

| Fault Code | Frequency | Fault Code | Frequency |
|---|---|---|---|
| A1 | 15 | B1C1 | 5 |
| A1B1 | 4 | C1 | 20 |
| A1C1 | 2 | C1D1 | 2 |
| A1D1 | 2 | | |

The number of faults observed was consistent with confirmed faults eliminating uncertainties.

To test the usability of SAFE, we implemented an example with 8 sensors measuring temperature. Table 2. shows the output from the SAFE framework.

**Table 2. SAFE Output**

| Time | Sensor | Value | Status | |
|---|---|---|---|---|
| 1:50 | S3 | 1.4 | Valid | Time |
| 1:50 | S4 | 1.5 | Valid | Stamp |
| 2:00 | S5 | 1.2 | Valid | = 2:00 pm |
| 2:00 | S6 | 1.3 | Valid | |
| 2:00 | S7 | 1.4 | Valid | Fused |
| 2:00 | S8 | 1.5 | Valid | Sensor |
| 1:20 | S1 | 1.2 | Stuck | Value |
| 1:20 | S2 | 1.3 | Stuck | = 1.3857 |

The output observed from SAFE was consistent with the inputs from functional sensors even when some sensors were forced to send wrong values. Faulty sensors were also detected.

## 5 Conclusion

In this paper we presented a modeling and simulation technique that anticipates the occurrence of faults/failures and is equipped with the relevant tools to deal with these faults promptly.

## REFERENCES

[1] S. Chaterji, P. Naghizadeh, M. A. Alam, S. Bagchi, M. Chiang, D. Corman, B. Henz, S. Jana, N. Li, S. Mou et al., "Resilient cyberphysical systems and their application drivers: A technology roadmap," arXiv:2001.00090, 2019.

[2] D. Niyonkuru and G. Wainer, "A Kernel for Embedded Systems Development and Simulation Using the Boost Library", Proceedings of the Symposium on Theory of Modeling and Simulation, April 3rd-6th, Pasadena, California, 2016.

[3] R. Rajkumar, I. Lee, L. Sha and J. Stankovic, "Cyber-physical systems: The next computing revolution", Proc. ACM 47th Design Automation Conference, June 13th-18th, Anaheim, California, USA, pp731-736, , 2010

[4] P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, and S. Achiche, "Design, modelling, simulation and integration of cyber physical systems: Methods and applications," Computers in Industry Vol.82 pp. 273-289, 2016