International Journal of Industrial Ergonomics 68 (2018) 89–100

Contents lists available at ScienceDirect

International Journal of Industrial Ergonomics

journal homepage: www.elsevier.com/locate/ergon

A simulation platform for human-machine interaction safety analysis of cyber-physical systems

Chin-Feng Fan[a,*], Ching-Chieh Chan[a], Hsiang-Yu Yu[a], Swu Yih[b]

[a] Computer Science and Engineering Department, Yuan-Ze Univeristy, Taiwan
[b] Department of Computer Science and Information Engineering, Chien Hsin University, Taiwan

ARTICLE INFO

Keywords:
Human-machine interaction (HMI)
Cyber-physical systems (CPSs)
Safety analysis
Accident scenarios
Model-level fault injection
Ptolemy II

ABSTRACT

Human-machine interaction (HMI) safety for cyber-physical systems (CPSs) is critical and its analysis is mandatory in many domains such as SCADA, autonomous cars, and medical devices. Generation of dynamic accident scenarios is the cornerstone of safety analysis. This paper presents a platform and associated methodology to effectively generate accident scenarios by modeling HMI errors using model-level fault injection, followed by simulation to produce dynamic evolutions of accident scenarios. An HMI fault tree template is proposed to guide the simulation. Results show that human mode confusion triggered by false displays may lead to severe accidents. Run-time assertions play a crucial role to detect HMI errors. A case study is described. The methodology expands Ptolemy II's modeling and simulation capabilities for CPS applications.

1. Introduction

A Cyber-physical system (CPS) (Rajkumar et al., 2010; Wiki, 2017) refers to a system that collaborates computing and networking to control physical processes through sensors, actuators and control algorithms. CPSs ranging from internet of things (IOTs) to industrial control systems are getting more and more common in contemporary daily life. Most of these CPSs are safety-critical; that is, system failures may cause life-threatening consequences. Thus, how to perform *safety analysis* (or interchangeably, *risk analysis*) for CPSs is critical. Moreover, new challenges are presented when considering safety of Human-Machine Interaction (HMI) processes of CPSs due to system heterogeneity and possibly complex cyber-physical-human interaction. Analysis of human factors or user misuse is mandatory in critical domains such as autonomous cars (US DOT, 2016), medical devices (US FDA, 2017), and aviation.

CPSs are frequently modeled by such tools as Metlab/Simulink or Ptolemy II (UC Berkeley, 2017; Ptolemaeus, 2014). Ptolemy II is a popular open-source design tool used to model and simulate complex, heterogeneous, and concurrent systems. However, neither an operator model nor safety analysis gets dedicated support from these tools. Besides, current CPS modeling mainly focuses on expected behavior; faults or errors can only be modeled on a case-by case basis. This research effort extends Ptolemy II by adding new fault generators and operator actors as well as a comprehensive fault injection method for HMI safety analysis.

We use “safety analysis” and “risk analysis” interchangeably. A safety/risk analysis aims to identify (i) potential accident scenarios, (ii) consequences of these scenarios, and (iii) their likelihood (Kaplan and Garrick, 1981). Among these tasks, the identification of accident scenarios is the most crucial and difficult, as elaborated in Siu's paper (Siu, 1994). This work aims to develop mechanisms to generate HMI dynamic accident scenarios systematically.

A CPS Safety Analysis and simulation Platform (*CP-SAP*) is constructed in this work. The platform *CP-SAP* introduces three new components on top of a regular Ptolemy II model: an interaction *fault tree* template, an *operator-decision* component, and a set of *fault generators*. Fault generators are systematically injected into any pair of interfaces between the cyber, physical, and human models to support simulation of accident scenarios.

Mitigation of hazards using run-time assertions is also discussed. An assertion is a predicate (a Boolean expression) expected to be true at that point in code. Our proposed assertion types include those checking values/ranges, checking temporal process-device dependencies, checking cyber-physical consistency, and checking global invariants.

A safety injection system is used as our case study. Observations from the case study show that assertions can be useful, and human mode confusion induced by false displays may lead to serious and unexpected accident sequences. The methodology can be easily adopted by design tools other than Ptolemy II. Moreover, the platform CP-SAP can be further extended to support CPS cybersecurity research in the future.

* Corresponding author.
E-mail addresses: csfanc@saturn.yzu.edu.tw (C.-F. Fan), ccchan.francis@gmail.com (C.-C. Chan), jumpingjump16@gmail.com (H.-Y. Yu), swuyih@uch.edu.tw (S. Yih).

https://doi.org/10.1016/j.ergon.2018.06.008
Received 7 January 2018; Received in revised form 21 June 2018; Accepted 27 June 2018

The contribution of this work is the effective methodology of HMI safety analysis. It injects possible HMI faults at model level, and then supports simulation of the model to produce dynamic accident scenario evolution over time with different event orders and timing. This is exactly that described by N. Siu as *dynamic risk analysis*, which is different from static event tree/fault tree methodology in that "the treatment of dynamic systems, i.e., systems whose responses to initial perturbations evolve over time as system components interact with each other and with the environment" (Siu, 1994).

The remainder of this paper is organized as follows: Section 2 briefly reviews background and related research followed by presentation of our approach in Section 3. After that, Section 4 describes a case study, a safety-injection system, including high-level modeling, performance, assertions, as well as a demonstration of realistic modeling of physical dynamics in lower-level details. Finally a summary and conclusion are presented in Section 5.

2. Background and related research

2.1. Accidents involving human-machine interaction

Human-machine interaction errors resulting in disasters did occur in the past. Failure modes of the following accidents are considered in this work. These failure modes are *misuse*, *false indication*, and *mode confusion* (a phenomenon in which the operator is confused about the current status of the automation system).

- A robot crashed a worker in Volkswagen production plant in Germany in June 2015 (Time, 2015) due to an operation error (or misuse) in which the worker was inside the cage while the robot was operating.
- The 2003 Northeast American 814 blackout (US-Canada Power System Outage Task Force, 2004) involved the cyber-physical-human interaction errors. When devices failed in the electrical grid, the computer display did not update the newest situation for two hours (i.e., false indication) due to a previous failure. Thus, the operator was misled by the display and ignored warning calls from outside. The event finally resulted in the biggest and costly blackout in American history.
- In the disastrous Three-mile island (TMI) accident (US NRC, 2014; IAEA, 2017), a stuck-open relief valve caused a small Loss-of-Coolant Accident (LOCA). The emergency core cooling system started up automatically. The actual state of the relief valve was open, but the control room indicator showed the valve was "closed". Furthermore, the indicator of water level of the pressurizer did not reflect actual water level (i.e., false indication) when there was coolant leakage through relief valve. The operator had mode confusion and stopped the injection of the cooling system manually. The second layer of protection systems automatically started soon; unfortunately, the operator stopped it again until more than 2 h later when the operator realized the problem.

2.2. Ptolemy II

Ptolemy II (UC Berkeley, 2017; Ptolemaeus, 2014), developed by UC Berkeley, is a widely-used open-source modeling and dynamic simulation tool for heterogeneous computation systems. Thus, it is particularly suitable for Cyber-Physical System (CPS) modeling and simulation.

Ptolemy II provides a hierarchical structure for heterogeneous computation in an actor-oriented fashion. Fig. 1 shows an example consisting of a *director* and several *actors*. Each actor has input and output ports to communicate with other actors. The director behaves as laws of physics to govern the interaction of actors. Ptolemy II has defined several types of directors, such as CT (continuous time), SDF (synchronous data flow), DE (discrete event), FSM(Finite State machines),etc. Both directors and actors can be modeled in a hierarchical fashion. However, current Ptolemy II does not provide dedicated components to support modeling of a human operator and that of safety analysis. We extend Ptolemy II by adding new actors to assist the modeling of the operator and fault generation.

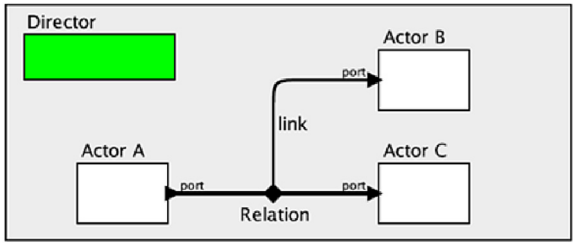

Fig. 1. Ptolemy II director and actors.

2.3. Safety analysis for CPSs

CPS safety-related issues have been investigated by academic and industrial communities. Research in safety verification involving Ptolemy II models has been done by L.C. Silva et al. (2015). They built a detailed insulin pump model in Ptolemy II and provided formal design verification including checking safety requirements using Simulink. However, safety verification is not equal to safety analysis because safety verification verifies against predefined safety requirements while safety analysis identifies possible life-threatening scenarios, their consequences and likelihood (Kaplan and Garrick, 1981).

Research in safety analysis for CPSs has also been conducted in academia. Banerjee's work (Banerjee, 2012) is the most detailed. He took an insulin injection system as the case study and modeled it in AADL (Architecture Analysis Design Language), and then performed safety analysis on the mathematical formulae that represented the dynamic interaction of the control software and the physical environment. However, failure modes considered by Banerjee mainly focused on communication failures. Besides, the human operator is not considered in his work.

Regarding human operators, human reliability analysis (HRA) has long been playing a critical role in certification for analog systems, such as that in nuclear power plants (Porthin, 2014). However, with the introduction of computer, the human-computer interaction (HCI) or digital human-system interface (HSI) becomes much more complex and needs new HRA guidelines (Boring, 2014).

As regards human-computer safety verification, Curzon et al. (2014) integrated PVS and Simulink to perform model-based formal verification of safety requirements involving human-computer interaction.

As to human-machine interaction safety analysis, our previous work investigated process interaction errors (Fan and Chen, 2000; Fan et al., 2011; Tseng and Fan, 2013), where processes refer to software, hardware, and human processes. This paper extends our previous work to cyber physical systems using Ptolemy II, which can provide more realistic modeling of physical dynamics.

3. Human-machine interaction safety analysis of CPSs

A Cyber-Physical system Safety Analysis and simulation Platform, called *CP-SAP*, shown in Fig. 2, is developed in this study to support augmenting a CPS model by an operator and interaction faults. CP-SAP contains the following three new components on top of a Ptolemy II model:

(1) A Cyber-Physical-Human (CPH) dynamic fault tree, called *CPH-fault tree*.
(2) A collection of fault generators.
(3) An operator module.