Safety Critical Systems in Medical Cyber-Physical Systems

Erdem Önal

1 Introduction

Medical devices are increasingly controlled by software systems that interact directly with patients. These Medical Cyber-Physical Systems (M-CPS) present safety challenges because software failures can lead to patient harm or death. This report examines three complementary safety methodologies for M-CPS and proposes a hybrid approach that integrates design time verification, runtime monitoring, and architectural standardization. The analysis uses a real FDA recall case of a medical ventilator to illustrate the critical gaps in current approaches and motivate the proposed solution.

2 Domain

The domain of this report is the safety and security of Medical Cyber-Physical Systems (M-CPS). In these systems, software components such as control modules, machine learning models, and medical applications control medical devices that directly interact with patients [1, 2]. Because these systems operate in direct contact with patients, they must be treated as safety-critical systems, since failures or unexpected behaviors can result in severe consequences such as patient harm or death [3].

3 Use Case Problem

A use case that shows the challenges in M-CPS safety is a U.S. Food and Drug Administration (FDA) recall of a medical ventilator, analyzed by Fu et al. [1]. The specific device, a Dräger Medical Evita V500 ventilator, was subject to a Class I recall[1] by the FDA. The failure was that the battery depleted much earlier than expected, even though the battery indicator showed a sufficient charge. Critically, the power fail alarm was not generated, creating a risk of patient injury or death from lack of oxygen. The root cause was not a hardware fault but an implicit design assumption. The system's software *Controller* was designed to calculate the remaining battery time and trigger an alarm based on this calculation. However, this algorithm was based on the unstated assumption that the ventilator would always operate in a temperature controlled environment where battery capacity is constant. When the device was used in lower temperatures, the battery's physical capacity was reduced. The *Controller*, being unaware of this environmental variable, miscalculated the remaining time and failed to send the alarm before total power loss occurred. The use case shows that M-CPS safety critically depends on explicitly modeling assumptions about the physical environment.

4 State of the Art

The state of the art for M-CPS safety can be analyzed across three distinct methodological layers. This transversal analysis moves from proactive design methodologies, to reactive operational methodologies, and finally to architectural methodologies.

[1]The FDA defines a Class I recall as "a situation in which there is a reasonable probability that the use of, or exposure to, a violative product will cause serious adverse health consequences or death."

4.1 Methodology 1: Proactive Design Time Verification

The first methodology, represented by Fu et al. [1], is proactive and proposes that safety must be engineered at the design stage. This methodology proceeds in four stages:

1. Problem Analysis: It takes actual failure scenarios as input, like the ventilator recall, to identify the root cause. For the ventilator, this was an implicit assumption about the physical environment.

2. Formal Modeling: It creates a mathematical model to explicitly define and compose these physical assumptions, such as temperature or humidity.

3. Model Integration: This mathematical model of assumption is then integrated into the main system design, for example, into its state diagrams.

4. Formal Verification: The final output is a new system design that is aware of its environment. This new design is formally verified using tools, such as UPPAAL, to prove it is safe even when environmental assumptions are violated.

4.2 Methodology 2: Reactive Runtime Monitoring

The second methodology, represented by Yasar and Alemzadeh [2], is reactive and addresses safety during the system's operation, especially in complex environments involving human operators. The process follows four main stages:

1. Data Input: The system takes kinematic data from the surgical robot's arms, such as position and velocity, as input.

2. Context Identification: A machine learning model identifies what the surgeon is currently doing. This provides the operational context.

3. Anomaly Detection: A second machine learning model checks if the robot's current movement is safe for that specific context. This is important because an action can be safe in one context but catastrophic in another.

4. Action: The output is a warning sent to the surgeon if an unsafe event is detected, caused by a system fault or a human error. This manages the complexity of systems with human operators.

4.3 Methodology 3: Architectural Standardization

The final methodology, from the white paper by Hatcliff et al. [3], addresses safety at the ecosystem level. This approach is structured into four stages:

1. Problem Analysis: It identifies the fundamental problem that the medical device industry is fragmented and lacks interoperability.

2. Obstacle Identification: It analyzes the key technical, regulatory, and ecosystem obstacles that prevent interoperability.

3. Principle and Platform Design: It proposes principles to overcome these obstacles and designs the concept of a Medical Application Platform or MAP.

4. Ecosystem Enablement: The output is an App Store model for healthcare. This allows certified devices from different manufacturers to connect to a common platform and run certified apps to coordinate safely, enabling a system of systems perspective.

4.4 Comparative Analysis and Modeling

The three methodologies approach M-CPS safety from different phases and philosophies. Methodology 1 is a proactive, design time approach. Methodology 2 is a reactive, runtime approach. Methodology 3 is a system level, architectural approach. Standard modeling languages can be used to visualize these methodologies.

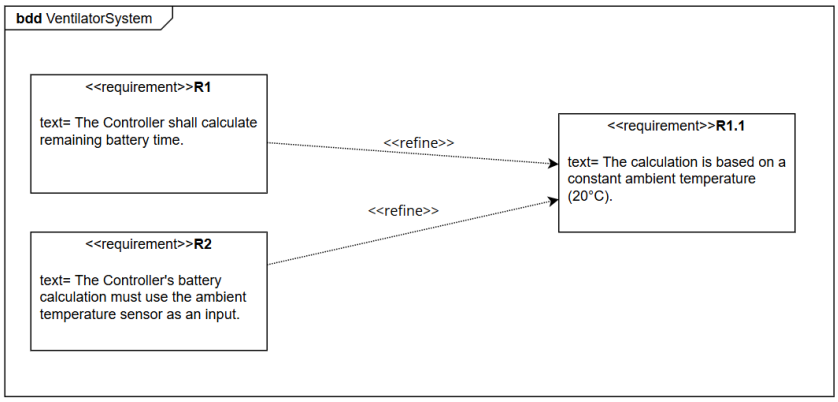

Figure 1: A SysML Requirement Diagram modeling the ventilator use case.

Figure 1 uses SysML to illustrate the core issue in Methodology 1. It shows how an overlooked implicit requirement (R1.1) shaped the main calculation task (R1), and how the updated requirement (R2) addresses the problem by making the environmental sensor an explicit input.

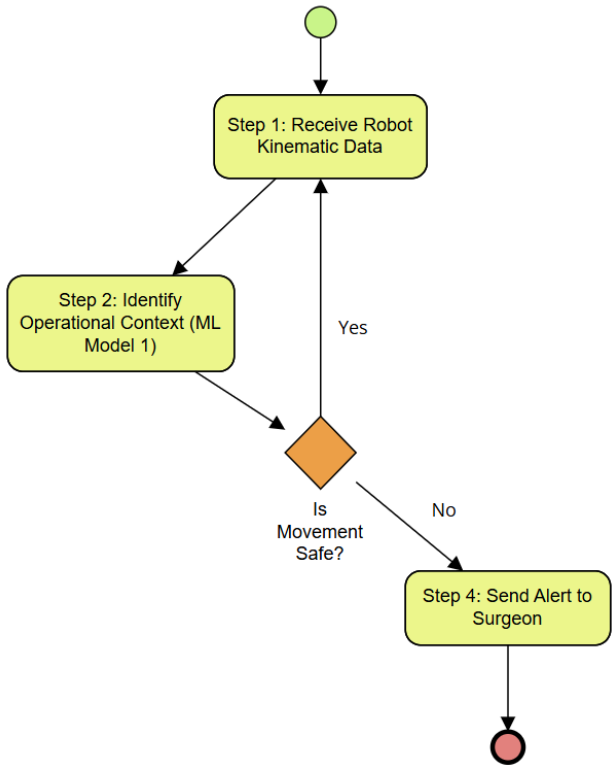

Figure 2: A BPMN Process Diagram illustrating the reactive loop of Methodology 2.

Figure 2 uses BPMN to map the reactive steps of Methodology 2. This diagram shows the continuous loop of data input, context identification, decision, and potential action.

These methodologies do not contradict each other. Instead, they support one another. The table below highlights their main differences.

Methodology	**Safety Philosophy**	**Focus Phase**	**Core Technique**
Assumption Modeling	Proactive (Preventive)	Design-Time	Formal Methods, Statecharts
Context-aware Monitoring	Reactive	Runtime	Machine Learning
Platform Standardization	System Level	Ecosystem/Architecture	System Architecture, Standards

Table 1: Comparison Table of the Three Safety Methodologies.

This analysis reveals a gap in the literature. Current methodologies do not provide an integrated framework that covers design time verification, runtime monitoring, and architectural coordination. This gap motivates the research question of this study.

5 Research Question

Based on the transversal analysis and the identified gap, the research question for this report is formulated as follows:

> How can an architectural methodology, such as the MAP framework [3], be extended to formally enforce proactive design time verification [1] and integrate reactive runtime monitoring [2] as verifiable components?

6 Research Plan

The proposed research plan is to develop a formal, hybrid methodology that integrates the three analyzed layers to address the research question. This requires combining the preventive capability of the design time approach with the operational safety mechanisms used during system execution.

6.1 Step 1: Hybrid Architectural Model

The first step is to formally model the proposed integration using a SysML Diagram (Figure 3). This model will show how the two main safety mechanisms, Design Time Verification and Runtime Monitoring, can be incorporated into the Medical Application Platform (MAP) architecture [3]. The integrated service must verify that new devices accepted by the MAP adhere to the corrected safety constraints. In addition to the structural view in Figure 3, Figure 4 provides an ARIS Event-driven Process Chain that describes how the proposed hybrid methodology works.

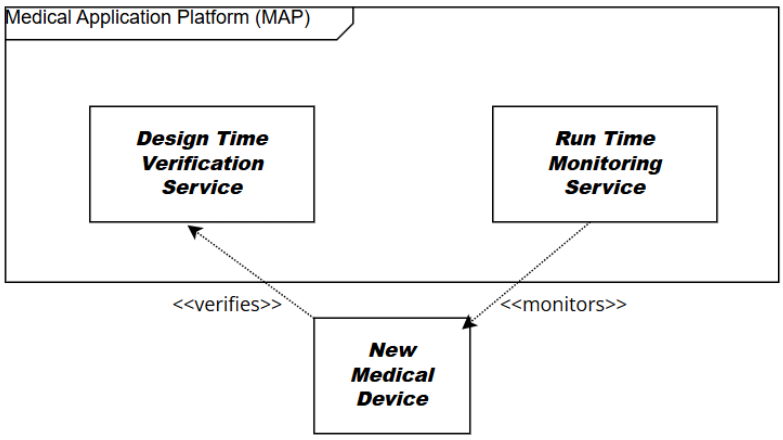

Figure 3: Hybrid Safety Architecture Model showing Design Time Verification and Runtime Monitoring within the MAP framework.

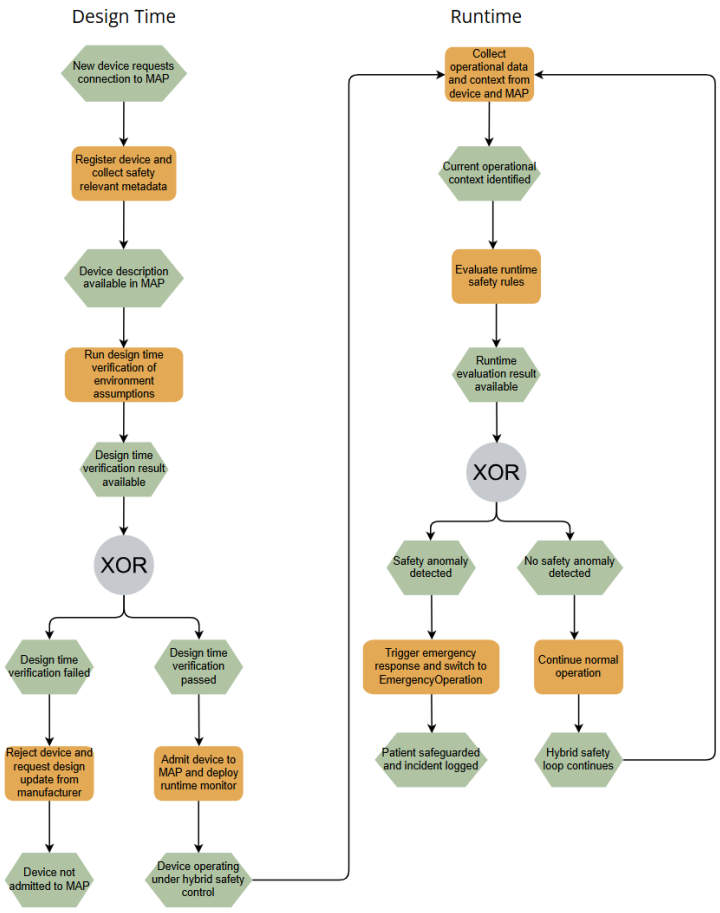

Figure 4: ARIS Event-driven Process Chain (EPC) for the Hybrid Methodology, showing the sequential integration of Design Time Verification and Runtime Monitoring within the MAP framework.

6.2 Step 2: Formalizing Methodology Elaboration

The next step is to describe how the parts work together within the hybrid architecture (Figure 4). This shows how the methodologies are integrated in two substeps:

1. The Design Time Verification service will be defined. The environmental assumptions from Fu et al. [1] will be written as safety rules. For example, `if temp < 0°C, then battery_capacity * 0.8`. These rules will use a formal notation such as Linear Temporal Logic (LTL). They will be verified with model checkers like UPPAAL. The resulting constraints become the safety rule set. New devices must satisfy this set before they can be used in the MAP. The Design Time Verification Passed gateway will enforce this requirement.

2. The Runtime Monitoring service will be formalized as part of the system. It may be implemented using ROS (Robot Operating System). The model output will not just raise a warning. Instead, it will be integrated as a guard condition in the operational state model. For example, the MAP central controller can switch states when needed. It could move from NormalOperation to EmergencyOperation. This switch would be triggered by the Safety Anomaly Detected gateway.

7 Evaluation Plan

The proposed hybrid methodology needs a clear way to check how well it performs.

7.1 Test Scenario

The evaluation will use the ventilator failure case as the primary test scenario. However, the scenario will be extended to include operational complexity. The simulation will introduce two concurrent failure modes:

1. Environmental Stress: The ambient temperature will be lowered to -10°C to trigger the battery capacity drop.

2. Operational Error: Simultaneously, a simulated human operator will fail to check the manual battery gauge.

7.2 Evaluation Metrics and Comparison

The proposed Hybrid Methodology will be compared with the earlier methods, based on two measures:

1. Safety (Detection Rate): The target is a combined detection rate above 95%. This builds on Methodology 2, where Yasar and Alemzadeh reported an F1 score of 88% for detecting surgical errors [2]. By adding Methodology 1 [1], which checks key environmental assumptions, the hybrid approach can catch problems that runtime monitoring alone may miss.

2. Timeliness (Response Time): The system targets a response time below 500 milliseconds. This is a safe and conservative bound. Methodology 2 showed an average reaction time of 57 milliseconds for human errors [2]. The goal is to keep latency low while also applying the MAP architectural checks [3].

Table 2 summarizes how the Hybrid Methodology is designed to address the limitations of the individual approaches, which is the core of this evaluation.

Safety Capability	**Methodology 1**	**Methodology 2**	**Hybrid Methodology**
Environmental Awareness	Yes	No	Yes
Human Error Detection	No	Yes	Yes
Multi-Device Coordination	No	No	Yes (via MAP)
Verification Method	Formal	ML	Both

Table 2: Comparison of Safety Coverage Across Methodologies.

This evaluation will check whether the Hybrid Methodology offers better safety coverage and quicker reactions than each method. Although the ventilator recall is used as a running example, the proposed hybrid methodology is intended to generalize to other Medical Cyber-Physical Systems, such as infusion pumps or robot-assisted surgical platforms, where environmental assumptions and human factors interact in similarly critical ways.

8 Ethical Aspects

Integrating autonomous safety mechanisms into medical systems creates critical ethical questions.

8.1 Liability and Accountability

If the hybrid system fails, it may be difficult to understand who is responsible. The issue could come from the device manufacturer, the platform provider, or another part of the system. Clear rules are needed to define who is accountable in such cases.

8.2 Data Privacy and Regulatory Compliance

The MAP architecture depends on regular data exchange between devices. Protecting this data and following privacy rules is an important concern. In addition, introducing this hybrid approach brings several regulatory issues. It requires clear steps for review and approval by agencies like the FDA, which are still adapting to software as a medical device.

8.3 Sustainable Development Goals

1. Goal 3 (Good Health and Well-being): The method verifies devices at design time and also monitors them during operation. The ventilator case shows that design assumptions about the environment can fail in real clinics. That failure can put patients at risk. The hybrid approach closes this gap and improves reliability in different clinical environments.

2. Goal 9 (Industry, Innovation and Infrastructure): MAP lets certified devices from different manufacturers coordinate safely. It uses standard interaction protocols that are manufacturer independent. This reduces fragmentation and improves interoperability. It also helps new medical technologies scale faster. Safety and regulatory requirements stay built into the system.

9 Conclusion

This report analysed three complementary approaches to safety in Medical Cyber-Physical Systems and identified a gap between design time verification, runtime monitoring, and architectural coordination. To address this gap, it proposed a hybrid methodology that embeds design time verification and context-aware runtime monitoring as safety services within a Medical Application Platform. The proposed architecture and ARIS EPC model show how these layers can be integrated. The evaluation plan based on the ventilator recall scenario explains how to check improvements in safety coverage and reaction time.

References

[1] Z. Fu, C. Guo, S. Ren, Y. Jiang, and L. Sha, "Modeling and Integrating Physical Environment Assumptions in Medical Cyber-Physical System Design," in *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Lausanne, Switzerland, 2017, pp. 1615-1618.

[2] M. S. Yasar and H. Alemzadeh, "Real-Time Context-aware Detection of Unsafe Events in Robot-Assisted Surgery," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Valencia, Spain, 2020, pp. 385-397, doi: 10.1109/DSN48063.2020.00054.

[3] J. Hatcliff et al., "Rationale and Architecture Principles for Medical Application Platforms," in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, Beijing, China, 2012, pp. 3-12, doi: 10.1109/ICCPS.2012.9.