Towards Safe and Effective Integration of Networked Medical Devices using Organ-based Semi-Autonomous Hierarchical Control [*]

Woochul Kang, PoLiang Wu, Lui Sha
Department of Computer Science
University of Illinois at Urbana-Champaign
{woochul,wu87,lrs}@illinois.edu

Richard B. Berlin Jr.
College of Medicine and Department of Computer Science
University of Illinois at Urbana-Champaign
Richard.Berlin@carle.com

Julian M. Goldman
Mass. General Hospital and CIMIT
jmgoldman@partners.org

Abstract

Leveraging connectivity and interoperability of medical devices promises a great benefit for patient safety and effectiveness of medical services. However, safety issues arising from coordination failures between networked medical devices pose a significant challenge to achieve such vision. In this paper, we propose an organ-based semi-autonomous hierarchical control structure as an architectural design principle to make integrated medical systems more resilient and effective against communication failures. The proposed design principle also enables the development of tools supporting rapid hierarchical composition of organ-based clusters and the verification of safety assertions. Our simulation study shows that our approach can provide the safety while minimally interrupting ongoing medical services in the face of network failures.

1 Introduction

In medical patient care environments, there are growing demands to leverage device connectivity and interoperability in order to improve the effectiveness of medical services and patient safety [17]. The increasing number of medical devices are meant to cooperatively automate medical workflows, implement smart alarms with integrated patient/treatment information and contexts, and provide safety interlocks that prevent human errors in dealing with networked devices. Recently, several initiatives have been launched, envisioning the efficient collaboration among medical devices while they reduce accidents caused by human errors [9]. One representative effort is ASTM *Integrated Clinical Environment* (ICE) standard [1], developed by the Medical Device Plug-and-Play (MD PnP) Interoperability program [8]. The ICE standard aims to provide standardized integration of data and devices to enable real-time control decision support and safety interlocks, thus ensuring patient safety. Several prototype designs and implementations of ICE have been reported [6, 13, 12].

However, one of the biggest challenges of the proposed ICE lies in guaranteeing the safety within the Plug-and-Play environment. If not properly designed, the introduction of the ICE supervisory control and the networking of medical devices could significantly increase the complexity of the whole system, tending to make the system more vulnerable to potential errors and safety hazards. Unlike standalone medical devices, whose safety is guaranteed by regulatory agencies such as FDA (Food and Drug Administration) after rigorous tests and verification, the ICE platform itself cannot be certified for its safety since it is, inevitably, a generic computing platform executing control functions of potentially arbitrary combinations of medical devices; the FDA is unable to certify computing platforms executing unknown combinations and configurations of devices at the time of certification. Therefore, the hardware and software architecture of ICE should be carefully designed to guarantee safety despite potential failures in underlying platforms and at any of the network communication points. For example, if a critical command from the ICE supervisory control is lost because of unreliable network connectivity, the ICE environment and design should have a mechanism to guarantee safety, such as device interlocking, despite the lost command.

In our previous work, we presented a safe supervisory framework, called *Network-Aware Supervisory Systems* (NASS) [12]. NASS is a first successful protocol that addresses open-loop safety issues of ICE. We demonstrated

[*] This research is supported in part by Grant Number U01EB012470 from the National Institute of Biomedical Imaging And Bioengineering, in part by NSF CNS 08-34709, and in part by ONR N00014-12-1-0046. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute Of Biomedical Imaging And Bioengineering or the National Institute of Health, the views of NSF, or the views of ONR.

NASS's open-loop safety mechanism under a simplified tracheal airway-laser surgery scenario. However, the effectiveness issue has yet to be addressed. By effectiveness, we mean supporting the execution of medical procedures as planned with little or no interruption. NASS assumes a two-level hierarchy ICE environment that consists of an ICE supervisor and all the devices. Under this communication architecture, when the network fails, no coordination between any devices are possible.

In this paper, we propose organ-based semi-autonomous and hierarchical control approach to support organ-based management of networked medical devices. Our goal is to provide architectural design principles and supporting mechanisms and tools that can be applied consistently to build safe and effective integrated networked medical devices. In our approach, devices are grouped into semi-autonomous clusters to support organ-specific homeostasis [1]. Each organ-based cluster is semi-autonomous since it can make independent control decisions under certain boundaries. For example, a ventilator device in the pulmonary cluster can adjust its oxygen pumping level within certain thresholds to maintain the homeostasis of the pulmonary/lung function of the human body. If homeostasis of an organ cannot be maintained by the local control of the cluster, an exception is raised to the higher ICE supervisory control device. The responsibility of the higher-level supervisory control is to handle such exceptional inter-organ cluster situations with help from health-care providers. This architectural principle makes the networked medical systems more safe and effective against failures since each cluster can still function and enforce per-cluster safety constraints even if the cluster loses connectivity and control from the remaining system.

Patient safety involves many dimensions, including potential drug interactions, allergies, and idiosyncratic reaction to given medical treatments. However, in this paper, we focus on safe and effective integration of networked medical devices. The contributions of this paper are as follows:

1) **Architectural design principles:** We provide design principles for safe and effective integration of networked medical devices. The design principles provide guidance in the choice of composition of devices, and the placement of functions, considering safety, scalability, and effectiveness.

2) **Supporting safety mechanism and tools:** We propose an open-loop safety mechanism to support safety against network failures. We also provide a toolkit in Simulink to support systematic application of the proposed design principles and open-loop safety mechanism. The toolkit provides device models, physiological organ models, and verification patterns for systematic composition, simulation, and verification of networked medical devices.

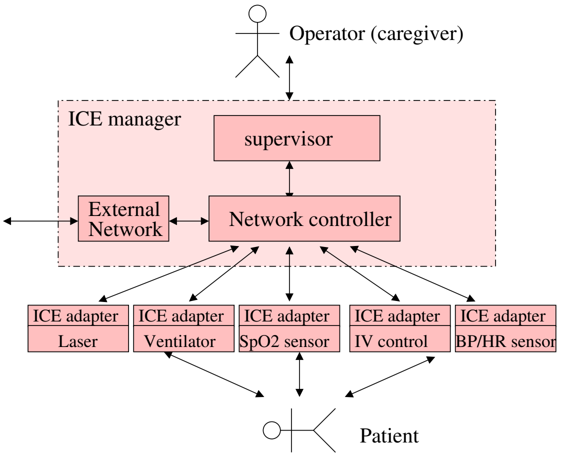


Figure 1. ICE with 5 devices plugged in

3) **Effectiveness evaluation:** Through a simulation study, the effectiveness of our approach is evaluated. We compare an architecture, which follows the proposed design principles, with baseline approaches. The evaluation results show that our approach can guarantee the safety while minimally interrupting the ongoing medical services in the face of network failures.

The rest of this paper is organized as follows. In Section 2, we present the background on ICE in the context of airway-laser surgery. In Section 3, we present the organ-based clustering design principle and its rationale. Additional design principles and supporting mechanisms are discussed in detail in Section 4. The supporting composition and verification tool is presented in Section 5. In Section 6, we discuss evaluation results. Related work is presented in Section 7, and Section 8 concludes the paper and discusses future work.

2 Clinical Background and Fault Model

Figure 1 shows the main components of the ICE (Integrated Clinical Environment) architecture proposed by the MD PnP project [1]. Medical devices are plugged into the *ICE manager* to build a virtualized integrated device. The ICE manager is a computer system that is responsible for supervising attached devices. The *supervisor* is the key component of the ICE manager, and automates medical workflows, implements smart alarms, and provides safety interlocks that prevent human errors. Hereafter, we use the term *supervisor* interchangeably with *ICE manager*, if not specified explicitly. The *adapter* at each medical device is a thin network device connecting the medical device to the ICE manager, either in wire or wireless. It should be noted that the architecture in Figure 1 is a logical view and does not impose a specific physical inter-networking topology.

[1] Human homeostasis is the ability or tendency of an organism or a cell to maintain internal equilibrium by adjusting its physiological processes.

Table 1. Potential safety hazards in airway laser surgery

Hazards	Safety constraints
Surgical fire	Laser device and ventilator should be interlocked. When the laser is enabled, oxygen supply should be blocked, vice versa.
Brain damage	Oxygen supply of ventilator cannot be blocked for more than 4.5 minutes.
Fluid overload	IV fluid infusion should be stopped, if changes in BP and HR are beyond thresholds for some time.

In this paper, we are only concerned about *coordination failures*, in which networked medical devices cannot provide intended functions and safety due to the loss of coordination between devices. In the ICE architecture, we identify two sources of a coordination failure: a failure of ICE supervisor and a communication channel failure. Since each medical device is certified by regulatory agencies, such as FDA, we do not consider failures of devices. In this paper, we treat a failure of ICE supervisor as a communication channel failure between the devices connected to the supervisory control. Byzantine faults are not considered in our fault model.

2.1 Airway Laser Surgery

A simplified version of airway-laser surgery was used in NASS to demonstrate its safety guarantees. Throughout this paper, we also extend the NASS scenario as a clinical example to illustrate the safety and effectiveness of the proposed approach.

An airway-laser surgery has the potential danger of an accidental fire if the laser is activated while high oxygen concentration is supplied by the ventilator. Whenever the laser is being activated, a human operator must block (or significantly reduce) the air path from the oxygen concentrate, first. However, in our simplified patient model, if oxygen is reduced for more than about 4.5 minutes and the blood oxygen saturation (SpO2) level is below 30%, the brain of the patient might be damaged. In traditional operating room environments, nurses and surgeons are supposed to be aware of such potential fire and low-oxygen problems [3]. Nevertheless, an unfortunate 100 fires are reported annually in the US due to the human errors during airway-laser surgeries.

For discussion in this paper, the patient undergoing tracheal airway-laser surgery is given as being elderly with a history of congestive heart failure (CHF). During airway-laser surgery, IV (intravenous) fluid, such as normal saline (NS), is commonly injected to deliver medications, or to prevent dehydration during the procedure. However, the volume of IV fluid should be carefully controlled since over-infusion of IV fluid can cause fluid retention, which can then trigger congestive heart failure, particularly in a patient with a history of heart disease. If the changes of blood pressure and heart rate are beyond thresholds set for a patient, then the IV fluid infusion should be stopped. This is just one of many common complications in ICU (intensive-care unit).

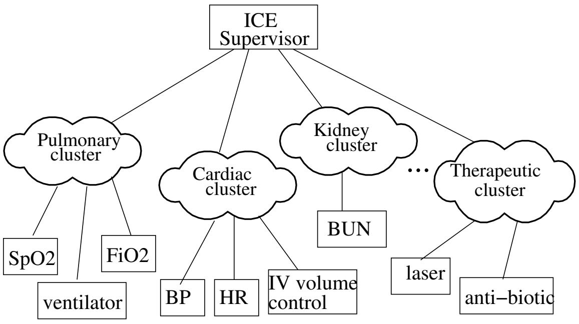


Figure 2. Organ-based clustering and hierarchical control structure.

Table 1 summarizes potential safety hazards that must be closely monitored by caregivers during the initial airway-laser surgery model, now with the addition of devices to monitor BP, HR, and IV fluid infusion. In such a highly error-prone traditional surgical environment, the supervisory control of ICE is expected to increase the safety. For example, the ICE supervisor can enforce the interlock between laser and ventilator automatically during surgery.

3 Organ-based Clustering Principle

The prevailing idea behind the previous prototypes and design of ICE architecture is the composition of an arbitrary set of medical devices into virtual medical devices. Hence, a flat structure, such as shown in Figure 1, has been assumed and it does not restrict any inter-connection and communication patterns. However, our *organ-based clustering* principle constrains the communication pattern to be clustered according to their physiological correlation and organ-centric human homeostasis. Figure 2 shows an example of an organ-based clustering and hierarchical control architecture. In this example, for instance, a ventilator, a blood oxygen saturation (SpO2) sensor, and a fraction of inspired oxygen (FiO2) sensor are all strongly tied to the respiratory system of the human body, and, hence they are grouped into the pulmonary cluster. Similarly, clusters for cardiac, kidney, therapeutic devices, and others might be identified.

The organ-based clustering approach has several advantages. From a medical perspective, human organs are loosely coupled to each other since each of them shows strong homeostatic behaviors. For example, the homeostasis of the cardiac system can maintain its blood pumping function against a modest fluid overdose and hardly affect pulmonary functions until the onset of congestive heart failure. From an engineering perspective, clustering is a primary way to control the complexity of large-scale distributed systems; clustering provides high scalability, fault isolation, timeliness, and the separation of concerns. With clustering, we can process tight and short timescale dynamics inside a cluster, while loose and long timescale dynamics between clusters are handled via inter-cluster coordination.

4 Organ-based Semi-Autonomous Hierarchical Control Architecture

In this section, we first describe a set of design principles that guide the development of *organ-based semi-autonomous and hierarchical ICE* implementation. We then describe supporting mechanisms and tools for the design principles.

4.1 Design Principles and Architecture

The organ-based clustering principle structures the design space for ICE implementation, but is not itself an architecture. The *organ-based semi-autonomous and hierarchical control architecture* is an ICE architecture based on the *organ-based clustering* principle and 3 additional *design principles* described here.

The first principle expresses the **Semi-Autonomous Hierarchical Control** property of the architecture: *Each organ-based cluster of devices has a semi-autonomous closed control loop supporting the homeostasis of the organ and the upper level supervisory control is only responsible for inter-cluster coordination.* This principle makes each cluster operate as a semi-autonomous unit, comprising sensing and treatment for the corresponding organ. For example, in the pulmonary cluster, when a patient is on the ventilator, an attending physician will set a lower-bound threshold on the blood oxygen level (SpO2) and upper-bound threshold on the ventilator control variables such as pressure, tidal volume, and oxygen concentration (FiO2). The ventilator controller can adjust the control variables to achieve target SpO2 level providing that no control thresholds are exceeded. We call this type of control *semi-autonomous* control because the pulmonary cluster can independently adjust control variables as long as no control thresholds are violated. Due to the homeostatic property of human physiology, the treatment at one cluster has weak influence on other clusters. This implies that organ-specific physiological model and corresponding controller can be designed in an independent and modular manner. Unlike homeostasis of separate organs, which can be managed by semi-autonomous computerized controllers, a diagnosis and treatment of symptoms involving multiple organs can hardly be automated by computers and it still belongs to the realm of human physicians. A situation requiring inter-organ coordination should be notified to the upper level supervisory control via exceptions. The only supported inter-organ coordination, in this paper, is the interlocking between the devices belonging to different clusters. Automated decision-making at the supervisory control is out of scope of this paper.

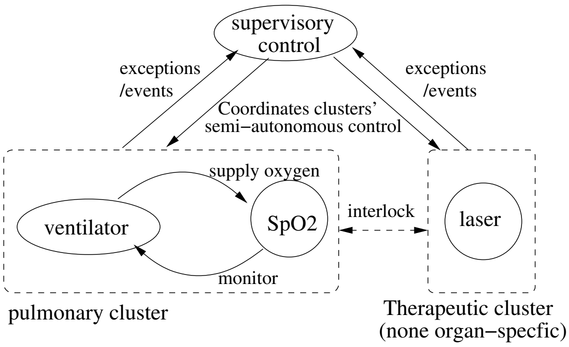


Figure 3. Semi-autonomous and hierarchical ICE architecture.

The second principle expresses the **Encapsulated Safety** property of an organ-based cluster: *Each cluster is responsible for specific safety constraints, and needs to guarantee the safety constraints even under network failures or loss of external control.* Specific safety constraints are encapsulated to a certain cluster and transparently handled by the cluster. For example, the brain damage from reduced oxygen supply is prevented by the pulmonary cluster by limiting the blocking time of oxygen supply. Since each organ-based cluster has autonomy, a cluster can function as far as the devices in the cluster can communicate and make coordinated decisions. For instance, even if the cardiac cluster is disconnected from the ICE supervisor, the cluster can maintain proper level of blood pressure and heart rates by adjusting the IV fluid volume within the thresholds. If the thresholds are violated when the cluster is disconnected from the upper level supervisor, the cluster should raise a local alarm to draw attention from surgeons.

The final principle expresses the **Cluster Header and Proxy Control** property: *For each cluster, a device is designated as a cluster header, which is computationally powerful enough to manage autonomy of the cluster. If there is no such device, a proxy control for the cluster should be placed at the upper level supervisory control device.* In actual placement of control functions, one of the devices should be a focal point, or cluster header, that interconnects devices in the cluster. For example, for a pulmonary cluster, a ventilator device can be designated as a cluster header. If the ventilator either does not support plug-and-play interface or is not computationally powerful enough, a proxy controller for the pulmonary cluster should be placed at the

Table 2. Safety levels of a cluster

SL_{sys}	Closed-loop control situations.
SL_{clst}	Cluster C is disconnected from the supervisor. But, C's devices can communicate locally.
SL_{vclst}	A network failure occurs inside cluster C. Per-cluster contingency plans are executed for a limited duration. After the time limit C's devices switch to default safety modes.
SL_{dev}	Each device in C is in default safe mode.

ICE supervisor, and the devices of the cluster need to be directly plugged into the ICE supervisor. This compromises safety since faults at the ICE supervisor might also affect the underlying clusters.

Figure 3 shows a hierarchical ICE architecture designed according to the design principles. In the pulmonary cluster, a semi-autonomous closed loop is formed between the ventilator and the monitored SpO2 level to aid the homeostasis of the organ. The safety constraints, such as preventing brain damage, are encapsulated in the pulmonary cluster. Once the cluster's safety constraints are verified, the cluster can be reused in various clinical scenarios in a modular manner. In the airway laser surgery scenario, the laser device in the therapeutic cluster is interlocked with the ventilator in the pulmonary cluster. Therefore, a coordination control logic should be placed at the ICE supervisor to ensure the interlocking. In this case, the pulmonary cluster's autonomy is compromised to support the inter-cluster interlocking property.

4.2 Closed-loop and Open-loop Safety

An architecture of networked medical devices designed according to the proposed design principles might include two types of control loops: intra-cluster semi-autonomous control loops and inter-cluster control loops. Intra-cluster control loops exist at each organ-based cluster and supports homeostasis of the corresponding organ. On the contrary, inter-cluster control loops exist at the ICE system level and coordinate multiple cooperating clusters [2]. If all network connections along a control loop are intact, we call this situation *closed-loop control*. Under *closed-loop control*, the devices in the control loop can communicate and make coordinated actions to effectively enforce safety constraints. This level of safety is called *closed-loop safety*. At the design time, the closed-loop safety should be checked by composing component devices' models and physiological models.

However, when any communication links along the control loops are broken, the safety assertions verified under closed-loop control situations are not valid anymore. When a control loop is broken due to network failures, this situation is called *open-loop control*. The level of safety provided under this situation is called *open-loop safety*. Open-loop control situations might happen either at inter-cluster control loops or intra-cluster control loops. In the proposed architecture, inter-cluster network failures do not affect semi-autonomous control loops embedded in each organ-based cluster. Hence, the clusters can still provide per-cluster safety. For instance, the pulmonary cluster can still prevent brain damage even under inter-cluster network failures. However, when an intra-cluster network failure occurs, the organ-based semi-autonomous control loop is broken and encapsulated per-cluster safety constraints cannot be guaranteed.

Table 2 summarizes the safety levels of a cluster under closed-loop and open-loop control situations. SL_{sys} represents closed loop control situations. The remaining three safety levels represent open-loop control situations under different conditions of network failures. Under SL_{dev} safety level, all devices stay in their own default safety modes; e.g, the laser device is disabled and the ventilator is activated. This is the most restricting open-loop control situation. While safe, it is also least effective since a surgery must be stopped. SL_{clst} is an open-loop safety level, in which inter-cluster network failures have occurred, but the cluster's devices can communicate with each other locally. Therefore, in SL_{clst} safety level, the encapsulated per-cluster safety constraints can be effectively provided according to the semi-autonomous clustering principle. SL_{vclst} is a transient open-loop safety level supporting encapsulated per-cluster safety constraints for a limited time despite intra-cluster network failures. For SL_{vclst}, contingency plans, which are discussed in Section 4.2.1, are exploited to make coordinated control actions in the face of intra-cluster network failures, instead of putting all devices of the cluster immediately into their respective per-device safety modes. Since medical services, e.g., airway-laser surgery, still can be performed at SL_{clst} and SL_{vclst} levels with little or no disruption, supporting two intermediate open-loop safety levels is a great advantage for improving effectiveness of medical services. In the meantime, the network failure might be resolved, allowing the system to recover to SL_{sys} or SL_{clst} levels.

4.2.1 Open-Loop Safety Mechanism

In our approach, the effectiveness under open-loop control situations is provided by contingency plans. Contingency plans consist of a set of coordinated future actions to be taken by each device in the event of network failures. Our system dynamically generates contingency plans from each device's contingency model. A contingency model M_d^C of

[2] As mentioned earlier, most inter-cluster coordination requires human physicians' involvement. In this paper, we only consider interlocking between clusters for inter-cluster coordinations.

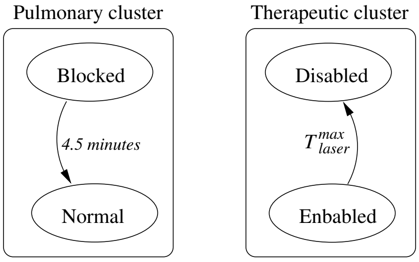


Figure 4. Contingency models of the clusters for inter-cluster interlocking.

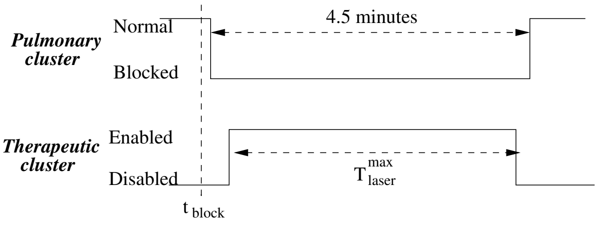


Figure 5. Contingency plans for interlocking.

a device d encodes the control behavior of the device under open-loop control situations. For example, when an intra-cluster network failure occurs, the contingency model of a ventilator automatically resumes the oxygen supply in 4.5 minutes to prevent brain damage.

In our architecture, each cluster header is responsible for generating instantiated contingency plans from component devices' contingency models. For instance, if a ventilator requests blocking the oxygen supply at time t_{blcok}, the cluster header can make a contingency plan for the ventilator, telling when the oxygen supply must resume. At the design time of a cluster, the contingency models and physiological models should be composed together and one should verify whether they can actually generate safe and non-conflicting contingency plans under open-loop control situations. In Section 5, we discuss tool support for the composition and verification of contingency models.

When two or more clusters must coordinate to guarantee certain safety assertions, a control logic is placed on the ICE supervisor. Currently, we only support the interlocking pattern for inter-cluster coordination. For inter-cluster interlocking, the same open-loop safety mechanism used for clusters can be used again. However, instead of dealing with device-specific models of underlying clusters, each cluster's contingency model related to the interlocking is expressed in an abstract manner. Figure 4 shows the contingency models of the pulmonary cluster and the therapeutic cluster. The pulmonary cluster is expressed as having two states, and all nominal behaviors of the ventilator and the SpO2 monitor are hidden in the *Normal state*. The laser device at the therapeutic cluster can stay at *Enabled* state as long as the pulmonary cluster is not in *Normal* state. While in *Enabled* state, the laser device can be both activated and deactivated by physicians. Figure 5 shows an example of contingency plans generated for the interlocked clusters. Once these contingency plans are approved and delivered to respective clusters at t_{block}, the laser device can be used during T^{max}_{laser} time period even if a network failure occurs. Hence, the airway laser surgery can still be performed without disruption within the time period. In the mean time, the network failures might be resolved, transparently masking network failures to operating physicians. Again, inter-cluster interlocking safety assertions should be verified at design time using tools, as discussed in Section 5.

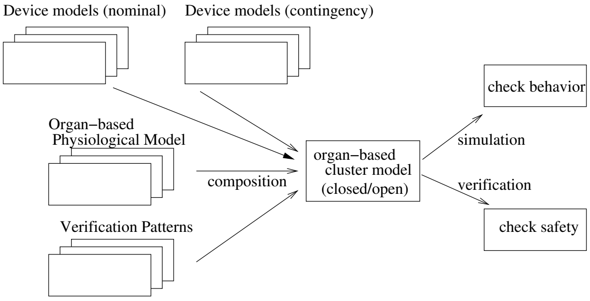


Figure 6. Tool support to simulate and verify organ-based clusters.

At runtime, whenever a device sends a request to change its state, its cluster header or the ICE supervisor for interlocked clusters needs to update the contingency plans for the devices before approving the request. This requires synchronous and transactional updates of devices' states for consistency. This issue was addressed in our earlier NASS protocol [12], and, hence, we omit the detailed mechanism in this paper. Interested readers are referred to [12].

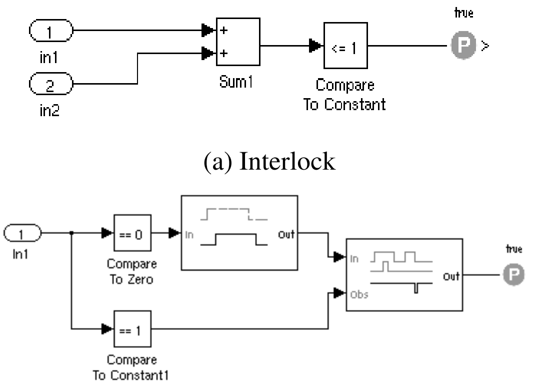
5 Tool Support for Design Principles

The design principles and supporting mechanisms are embodied within the simulation and verification toolkit, which is being implemented in MATLAB Simulink. The toolkit enables clinical engineers to rapidly compose organ-based clusters and hierarchical control structures to verify both closed-loop and open-loop safety.

5.1 Components

As shown in Figure 6, the toolkit has 3 reusable components: a set of *medical device/cluster models*, a set of *organ-specific physiological models*, and *verification patterns*.

The *device models* are executable models that simulate the control actions of devices. Each device has two control models: one for modeling nominal control behavior under closed-loop control, and the other for modeling contingency control behavior under open-loop control. As discussed in Section 4.2.1, the contingency behavior model of a device is used by its cluster header or the ICE supervisor to generate contingency plans at runtime. Medical device manufacturers are supposed to provide device models. Currently, all devices models are modeled using Stateflow charts of Simulink.



(a) Interlock

(b) Time bound (maximum)

Figure 7. Verification patterns.

Holistic modeling of human physiology is a challenging task, if not impossible. In our organ-based clustering approach, only organ-specific human physiology needs to be modeled. For example, the saturated oxygen (SpO2) level of a human body can be modeled as follows:

$$SpO2(k+1) = SpO2(k) + (K_s + K_c) \times \delta t, \qquad (1)$$

where K_s and K_c are the oxygen supply rate from the ventilator [3] and the metabolic oxygen consumption rate of the human body, respectively. Physiological models are used to check the behavior of the clusters through simulations. Further, the physiological models are also used by organ-based clusters to generate contingency plans at runtime. Unlike actual measurements from sensors, physiological models have an effective time bound since the discrepancy between the model and real physiological state grows over time if not updated regularly. That is the reason why the safety level SL_{vclst} has a limited time bound. If a network failure persists, all devices should be put into SL_{dev} safety level before the time bound. Clinical engineers are supposed to provide these organ-based physiological models. Currently, physiological models are modeled using Simulink Stateflow charts and discrete transfer functions.

For formal specification of safety constraints, or assertions, we provide verification patterns. Figure 7 shows two of them: *interlock* pattern and *time bound (maximum)* pattern. The *interlock* pattern takes two signals, *in1* and *in2*, and verifies whether at most one of the signals is true. For *time bound pattern (maximum)* pattern, a signal, *in1* is allowed to stay at one state for at most a specified time period. In the airway laser surgery scenario, the *interlock* pattern can be used to specify the interlocking between the laser device and the ventilator. The *time bound (maximum)* pattern can be used to specify the maximum time bound for threshold violation in the cardiac cluster and for oxygen blockage time in the pulmonary cluster. The verification patterns are being implemented using Simulink Design Verifier toolbox.

5.2 Verification of Safety

Once a cluster and its encapsulated safety constraints are identified, its device models, physiological models, and instantiated verification patterns are composed into a cluster for verification. A cluster's safety assertions should be verified both under closed-loop and open-loop conditions.

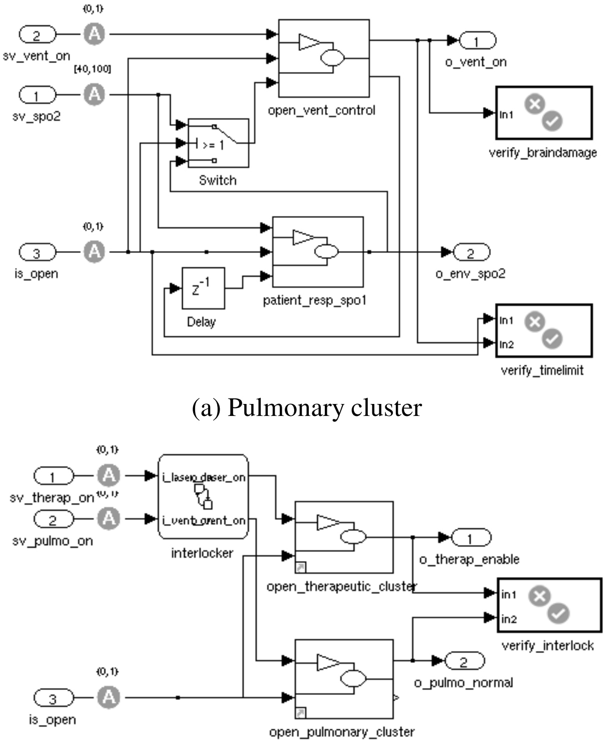
For closed-loop safety of a cluster, nominal behavior models of devices are composed together with physiological models to verify encapsulated safety constraints. For instance, the pulmonary cluster should guarantee that brain damage does not happen under closed-loop control situations. Due to space limitation, we omit the details of the safety verification for closed-loop control systems. Interested readers are referred to [5].

As discussed in Section 4.2.1, a cluster header exploits contingency models of its component devices and physiological models to dynamically generate per-cluster contingency plans for open-loop safety at runtime. Therefore, these models should be composed together and one should verify whether safety assertions can be guaranteed under open-loop control situations. For the airway-laser surgery, the following things are verified for safety constraints listed in Table 1.

Pulmonary cluster: Figure 8-(a) shows the composed pulmonary cluster to verify the safety assertions under open-loop situations. In the figure, the 3rd input *is_open* is the trigger of an intra-cluster communication failure. For the verification, all inputs of the composed system are left open and the embedded model checker of Simulink checks all possible state space. Safety constraints are shown on the right side in Figure 8-(a). The safety assertion specified in *verify_braindamage* describes that the oxygen supply of the ventilator should never be blocked for more than the specified time bound. This safety assertion is described using the *time bound(maximum)* verification pattern. The only input to the safety assertion is the state of the ventilator.

Cardiac cluster: The prevention of fluid overload is achieved by detecting the duration of blood pressure and heart rate threshold violations set by the attending physicians. This safety assertion can be modeled using the *time*

[3] This is an abstraction that models the effects of the settings of FiO2 partial pressure, tidal volume, and breathing rate.



(a) Pulmonary cluster

(b) Inter-cluster interlocking

Figure 8. Composition and verification for open-loop safety.

bound(maximum) verification pattern. Because of the simplicity of the solution and lack of space, we do not show the verification in this paper.

Inter-cluster interlocking: For hierarchical composition of clusters, the same composition and verification method used for individual clusters can be used. The only difference is that the inter-cluster composition takes high-level cluster models, not detailed device models, as inputs for the composition and verification. High-level cluster models, such as shown in Figure 4, are constructed according to the verified properties of the clusters. Figure 8-(b) shows that the high-level cluster models of the pulmonary cluster and the therapeutic cluster are composed for the verification of the interlocking safety assertion. The *verify_interlock* block in Figure 8-(b) takes the states of the pulmonary cluster and the therapeutic cluster as inputs to check if the interlocking property is guaranteed.

6 Evaluation of Effectiveness

In this section, we evaluate the effectiveness of the proposed design principles through a simulation study.

At SL_{vclst} and SL_{dev} safety levels, the effectiveness of the medical service is limited because the devices should either strictly follow contingency plans or stay at their default safety modes. In particular, under SL_{dev} safety level, minimal safety is guaranteed, but medical procedures should be stopped, significantly reducing the effectiveness of the medical services. Therefore, we quantify the ineffectiveness of networked medical devices as follows:

Table 3. Evaluated approaches

SYS_{clstr}	Organ-based semi-autonomous clustering, and contingency plans are supported
SYS_{nass}	No clustering, but supports contingency plans for open-loop safety
SYS_{base}	Neither supports organ-based clustering nor open-loop contingency plans

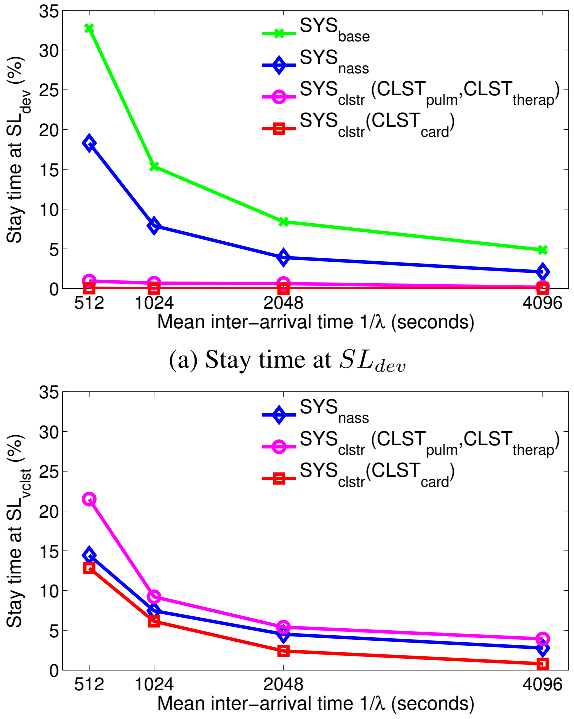
$$\frac{\text{stay time at } SL_{dev} \text{ level}}{\text{total execution time}} \times 100(\%), \qquad (2)$$

To determine the effectiveness of the proposed design principles, we compare the behavior of our approach with baseline approaches under various conditions, where a set of parameters have been varied. We consider 3 approaches shown in Table 3. In SYS_{base}, all medical devices are directly plugged into a centralized supervisory device. Under a communication failure at one or more devices, all plugged-in devices transit into default per-device safety modes, or SL_{dev}, to prevent potential safety hazards. This is a most straightforward implementation of the ICE standard. SYS_{nass} is similar to SYS_{base}, but devices have contingency plans for communication failures. This approach is similar to our previous NASS prototype [12]. Finally, SYS_{clstr} is our approach, which has organ-based semi-autonomous clusters and the hierarchical control structure. Each cluster has per-cluster contingency plans for open-loop safety.

We assume the airway-laser surgery scenario introduced in Section 2. For SYS_{clstr}, 3 clusters are assumed: pulmonary cluster $CLST_{pulm} = \{ventilator, SpO2\ sensor\}$, cardiac cluster $CLST_{card} = \{IV\ control, BP/HR\ monitor\}$, and therapeutic cluster $CLST_{therap} = \{laser\}$. The pulmonary cluster and the therapeutic cluster are interlocked, and, hence, they transit to SL_{vclst} safety level together for open-loop safety. The contingency plans are assumed valid for 60 seconds for the pulmonary and therapeutic clusters. For the cardiac cluster, the contingency plans are assumed valid for 30 seconds. After those time bounds, the devices in the clusters, or ICE supervisor, transit into default per-device safety mode. For SYS_{nass}, the validity of contingency plans is determined by the minimum, which is the cardiac function, and, hence, its contingency plans are valid for 30 seconds.

During the simulation, we introduce failures at communication links. The failures are independent and its inter-arrival time follows an exponential distribution. Each failure is assumed to persist for a uniformly distributed time period between 10 and 110 seconds. After the recovery of a link, it takes an additional 5 seconds to make devices' states consistent.

6.1 Results



(a) Stay time at SL_{dev}

(b) Stay time at SL_{vclst}

Figure 9. Effectiveness.

Figure 9 shows the results when the mean inter-arrival time $1/\lambda$ of link failures is varied from 512 seconds to 4096 seconds. Figure 9-(a) shows that SYS_{clstr} is significantly more effective than other approaches. For instance, when $1/\lambda$ is 512 seconds, SYS_{base} and SYS_{nass} stay at SL_{dev} for about 33% and 18% of the total time, respectively, due to the communication failures. This implies that the surgery should be stopped for a time, significantly reducing the effectiveness. In contrast, SYS_{clstr} stays at SL_{dev} less than 1% of the total time. Since the pulmonary cluster and the therapeutic clusters are interlocked, they stay at the same safety levels. However, the cardiac cluster is independent from other clusters, and, hence, the cluster has less than 0.04% stay time at SL_{dev} safety level.

As the mean inter-arrival time of link failures increases, the effectiveness improves in all approaches. However, SYS_{base} and SYS_{nass} still have high percentages of stay time at SL_{dev}. For instance, when $1/\lambda$ is 4096 seconds, the airway laser surgery should be stopped for 4.87% and 2.10% of the total time in SYS_{base} and SYS_{nass}, respectively. In contrast, for SYS_{clstr}, the surgery is stopped only for 0.17% of the total time.

Figure 9-(b) shows how long each approach stays at SL_{vclst} safety level. Under SL_{vclst}, respective contingency plans are executed to provide safety against link failures, but the surgery is not stopped for the duration of the contingency plans. If link failures are not resolved before the expiration of the contingency plans, the system must transit into SL_{dev} safety level, reducing the effectiveness. Figure 9-(b) shows that the stay times at SL_{vclst} of SYS_{clstr} and SYS_{nass} are not significantly different. However, as shown in Figure 9-(a), the stay time at SL_{dev} of SYS_{clstr} is significantly smaller than SYS_{nass}. This demonstrates that SYS_{clstr} masks link failures more effectively than SYS_{nass}. Since each cluster of SYS_{clstr} is more independent from failures at other clusters, it can recover to closed-loop safety levels as soon as its local link failures are resolved. In contrast, SYS_{nass} cannot switch to closed-loop safety levels until all communication links are recovered.

7 Related Work

Medical Device Plug-and-Play (MD PnP) aims to improve the flexibility and interoperability of medical systems [8], and our work is part of that ongoing effort. Software architectures for communications in medical plug-and-play systems have been explored by King et. al. [13] using publish-subscribe middleware. A meta-model for describing medical devices has been proposed by Hofmann to support the interoperability of legacy devices [10]. Until now, however, much of the work for medical device interoperability has focused on establishing dynamic connectivity of devices, device-to-device synchronization, and ensuring fair access to a communication medium. To the authors' best knowledge, our current work along with our previous work [12] is the first addressing the open-loop safety issues for collaborating medical devices.

Medical device safety has been a prevalent issue dating back to the infamous incidents of Therac 25 radiation therapy machines [15]. Some mechanisms are used to improve the accuracy and safety of the systems, such as fuzzy logic [19], and information technology [7]. Arney *et. al* proposed a closed-loop control design for patients' safety using patient-controlled analgesia (PCA) infusion pumps. In our earlier work [12], we proposed *Network-Aware Supervisory System* (NASS) framework [12]. NASS guarantees open-loop safety by dynamically generating contingency plans for plugged-in devices. In this work, we extend NASS to support organ-based semi-autonomous clustering and hierarchical control structure of networked medical devices. Through a simulation study, we showed that this architectural support can make the implemented ICE systems more effective. Further, our approach supports model-based engineering using supporting tools. The tools enable rapid composition and verification of the proposed organ-based hierarchical ICE architecture.

Formal verification and validation of medical devices is a critical issue for patient safety [14]. For one thing, the U.S. Food and Drug Administration (FDA) enforces a strict certification procedure for medical device approval [16]. For another, formal methods are widely used in specifying and verifying medical devices to improve safety and reliability [2, 18, 11, 4]. However, much of the previous work focused on individual devices and did not address the safety verification arising from coordination failures in networked medical devices. In the current work, we aim to improve the coordination safety and effectiveness for networked medical devices. The tools developed in Simulink enable modular verification of the proposed organ-based architecture. In addition, once organ-based clusters are verified, they can be used as building blocks for hierarchical composition of networked medical devices.

8 Conclusions and Future Work

In this paper, we proposed design principles and supporting tools for safe and effective integration of networked medical devices. The proposed organ-based semi-autonomous and hierarchical control principle exploits properties of human physiology to make an integrated medical system more effective and resilient against communication failures. The modular structure of the proposed architecture enables rapid composition of organ-based clusters and the hierarchical verification of safety assertions. Through a simulation study, we showed that the proposed semi-autonomous hierarchical control structure can support encapsulated safety properties while incurring significantly smaller number of interruptions to ongoing medical services, compared to baseline approaches.

Currently, our work only supports the interlocking pattern for the hierarchical coordination of clusters. In the future, we plan to extend our work to support more inter-cluster coordination patterns. We also plan to apply the proposed design principles to our NASS prototype.

References

[1] ASTM F2761-09 medical devices and medical systems - essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) - part 1: General requirements and conceptual model, 2009.

[2] R. Alur, D. Arney, E. Gunter, I. Lee, J. Lee, W. Nam, F. Pearce, S. Van Albert, and J. Zhou. Formal specifications and analysis of the computer-assisted resuscitation algorithm (CARA) infusion pump control system. *International Journal on Software Tools for Technology Transfer (STTT)*, 5(4):308–319, 2004.

[3] S. D. amd Loeb RG. Laser surgery and fire hazards in ear, nose, and throat surgeries. *Anesthesiol Clin*, 28(3):485–496, September 2010.

[4] D. Arney and et. al. Formal methods based development of a PCA infusion pump reference model: Generic infusion pump (GIP) project. *HCMDSS-MD PNP*, 2007.

[5] D. Arney, M. Pajic, J. Goldman, I. Lee, R. Mangharam, and O. Sokolsky. Toward patient safety in closed-loop medical device systems. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, pages 139–148. ACM, 2010.

[6] D. Arney, M. Pajic, J. M. Goldman, I. Lee, R. Mangharam, and O. Sokolsky. Toward patient safety in closed-loop medical device systems. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '10, pages 139–148, New York, NY, USA, 2010. ACM.

[7] D. Bates and A. Gawande. Improving safety with information technology. *New England Journal of Medicine*, 348(25):2526–2534, 2003.

[8] J. Goldman, S. Whitehead, S. Weininger, and M. Rockville. Eliciting clinical requirements for the medical device plug-and-play (MD PnP) interoperability program. *Anesthesia & Analgesia*, 102:S1–54, 2006.

[9] High Confidence Software and Systems Coordinating Group. High-Confidence Medical Devices: Cyber-Physical Systems for 21st Century Health Care. A Research and Development Needs Report, NITRD, Feb. 2009.

[10] R. M. Hofmann. Modeling medical devices for plug-and-play interoperability. Master's thesis, MIT, 2007.

[11] R. Jetley, S. Purushothaman Iyer, and P. Jones. A formal methods approach to medical device review. *Computer*, 39(4):61–67, 2006.

[12] C. Kim, M. Sun, S. Mohan, L. Sha, and T. F. Abdelzaher. A framework for the safe interoperability of medical devices in the presence of connection failures. In *ACM/IEEE ICCPS*, 2010.

[13] A. King, S. Procter, D. Andresen, J. Hatcliff, S. Warren, W. Spees, R. Jetley, P. Jones, and S. Weininger. An open test bed for medical device integration and coordination. In *ICSE Companion*, pages 141–151. IEEE, 2009.

[14] I. Lee, G. Pappas, R. Cleaveland, J. Hatcliff, B. Krogh, P. Lee, H. Rubin, and L. Sha. High-confidence medical device software and systems. *Computer*, 39(4):33–38, 2006.

[15] N. Leveson and C. Turner. An Investigation of the Therac-25 Accidents. *IEEE Computer*, 26(7):18–41, July 1993.

[16] W. Maisel. Medical device regulation: an introduction for the practicing physician. *Ann Intern Med*, 140(4):296–302, 2004.

[17] MD PnP. Medical device "plug-and-play" interoperability progam. http://mdpnp.org, 2012.

[18] A. Ray and R. Cleaveland. Unit verification: the cara experience. *International Journal on Software Tools for Technology Transfer (STTT)*, 5(4):351–369, 2004.

[19] N. Stevens, A. Giannareas, V. Kern, A. Viesca, M. Fortino-Mullen, A. King, and I. Lee. Smart alarms: multivariate medical alarm integration for post CABG surgery patients. In *Proceedings of the 2nd ACM SIGHIT symposium on International health informatics*, pages 533–542. ACM, 2012.