Toward a Safe and Secure Medical Internet of Things

www.mdpnp.org

David Arney
Lead Engineer, MD PnP Interoperability Program
Massachusetts General Hospital

Julian Goldman, MD
Medical Director of Biomedical Engineering for Partners HealthCare System
Director, MD PnP Interoperability Program
Massachusetts General Hospital, Harvard Medical School
JMGOLDMAN@mgh.harvard.edu

1. INTRODUCTION

The landscape of modern medicine is dramatically changing with the advent of networked medical devices. This change brings the promise and the challenge of next-generation integrated medical systems that will interoperate efficiently, safely and securely. It is anticipated that it will significantly lower the rates of preventable medical errors, now estimated to be as high as the third leading cause of death in the U.S. [1]; and by providing improved patient outcome at lower costs [2]. Such improvements include, but are not limited to, support for real-time clinical decision support and automatic diagnosis, real-time checking of adverse reactions to medications, reduced false alarms and physiologic closed-loop control systems [5][6].

The grand vision of the Medical Internet of Things (MIoT) is to enable the deployment of patient-centric and context-aware networked medical systems in all care environments, ranging from homes and general hospital floors to operating rooms and intensive care units. Heterogeneous devices in each care environment would effectively share data – efficiently, safely and securely to minimize preventable errors that are often induced unknowingly by human operators. As medical devices move between different care environments or from patient to patient, they would securely discover other devices that they need to interoperate with, and then verify and execute safe, authorized and compliant operational profiles. The key to realizing this vision is coming up with standardized architectures that balance utility, reliability and safety requirements with those of security and privacy, and providing this information as a roadmap.

The Integrated Clinical Environment (ICE) framework, as defined by the ASTM F2761-09 standard [1], is a significant step toward enabling this interoperable MIoT vision. Most recently, with support from the US Government, we have been making advances to integrate security into ICE. Security considerations for interconnected and dynamically composable medical systems are critical not only because laws such as the Health Insurance Portability and Accountability Act (HIPAA) [4] mandate it, but also because security attacks can have serious safety consequences for patients. As these medical devices will be brought together and mixed/matched in an ad hoc fashion to serve the needs of a given patient (dynamically composed systems), additional security mechanisms will be required. They will need to support automatic verification that the system components are being used as intended in the clinical context, that the components are authentic and authorized for use in that environment, that they have been approved by the hospital's biomedical engineering staff and that they meet regulatory safety and effectiveness requirements.

As far as medical device communications is concerned, few of the existing or proposed standards for dynamically composed and interoperable medical devices and information systems include sufficiently comprehensive or flexible security mechanisms to meet current and future safety needs. There are significant gaps between required security properties and those that can be fulfilled even by combinations of currently standardized protocols [2]. Safety considerations in these standardization efforts are effectively incomplete due to a lack of appropriate security analysis.

Regulators are also noting the importance of incorporating security for safety and privacy in the medical domain. The FDA is calling for medical device manufacturers to address cyber-security issues for the *entire* lifecycle of the device: from the initial design phase through deployment and end-of-life [8][9]. Although these calls are in the form of draft guidelines for ensuring device security and interoperability, there is evidence that the FDA intends to use them as a basis for clearing medical device submissions [26]. This seems to be addressing the traditional lack of incentive for medical device manufacturers to incorporate necessary security mechanisms in their products for fear of complicating regulatory approval [27].

In this paper, we present recent research on protecting the communications within ICE based on the fine-grained security mechanisms provided by the OMG Data Distribution Service (DDS) standard. In Section 2, we provide a background on ICE and the components that comprise ICE systems. We provide an overview of the DDS standard suite, which forms the connectivity platform of OpenICE [4]; OpenICE is the ICE reference implementation. We also briefly introduce the DDS Security architecture for granularly protecting DDS-based communications. Sections 3 and 4 go over our analysis, developed prototypes and results.

Real-Time Innovations (RTI) and the Medial Device Plug-and-Play (MD PnP) Program at the Massachusetts General Hospital have collaborated on this research. We are planning on applying our findings to the Industrial Internet Consortium's Connected Care Testbed.

2. Background

2.1 Background on Integrated Clinical Environments (ICE)

The ICE framework, as defined by the ASTM F2761-09 standard [1] provides an approach for integrating heterogeneous medical devices and coordinating their activities to automate clinical workflows. From a high-level perspective, the idea behind ICE is to allow medical devices that conform to the ICE standard, either natively or using an after-market adapter, to interoperate with other ICE-compliant devices regardless of manufacturer. A similar paradigm has existed for many years in the personal computing domain, leading to an explosion of devices supporting WiFi, USB or Bluetooth standards. A similar approach in the medical domain, if done correctly, would enable dramatic improvements to patient safety. Known examples include patient transfers from the Operating Room (OR) to Intensive Care Units (ICU) or reducing false alarms in Patient-Controlled Analgesia (PCA) systems. In both of these examples, cross-vendor inter-device communications significantly reduces preventable medical errors [2].

Figure 1 depicts the general architecture of ICE and how it maps to the equipment of a test-bed setup at the MD PnP Interoperability Lab.

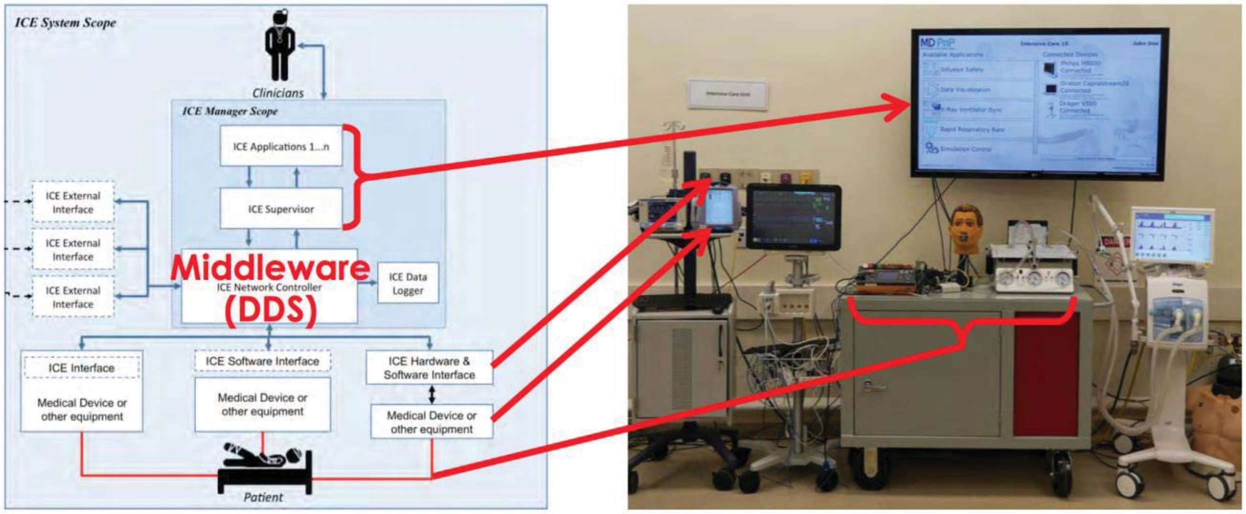


Figure 1. General architecture of ICE and an instantiation of it in a test setup at MD PnP Lab.

The *ICE Network Controller* is essentially a high-assurance middleware that forwards data or commands to or from ICE applications and devices, ensures communication quality-of-service and is agnostic as to the intended use of the clinical apps that it supports. It also manages the discovery and connection protocol for devices that wish to connect to the system. Given its critical communicatory role in ICE, having high-performance and context-aware security support in the network controller is paramount. The major functional security requirements for the network controller include: i) having authentication mechanisms for validating the identity of devices and apps, vouching for their provenance and ICE compliance, ii) having flexible yet easy-to-use mechanisms for defining and enforcing access control policies for various ICE configurations in different care environments, iii) having a mechanism for secure device and app discovery, iv) having a secure auditing mechanism and v) having mechanisms to guarantee the integrity, freshness and confidentiality of data. Note that the functional requirement should be met via a solution that has minimal negative impact on non-functional requirements such as performance, availability, robustness, and ease of use for clinicians and developers.

The *ICE Supervisor* provides separation/isolation-kernel-like data partitioning and time partitioning. It makes sure the information cannot inadvertently leak between apps and apps cannot inadvertently interfere with one another. It provides real-time scheduling guarantees that the computation in one app cannot cause the performance of another to degrade or fail. It also provides a console that allows a clinician to launch apps, monitor their progress and provide user-input during app execution. The ICE Network Controller and Supervisor may be incorporated together and deployed as a standalone ICE Manager.

ICE Applications are programs that accomplish a clinical objective by interacting with one or more devices attached to the network controller. As each app executes in the supervisor, it defines the intended use of the current ICE configuration. An important safety- and security-related concept is that ICE medical devices never interact directly with each other; all interaction is coordinated and controlled via the ICE apps. It is crucial that ICE apps exactly correspond to the specified task they were designed for.

The *ICE Data Logger* is dedicated to logging communication and other important events within the Network Controller and Supervisor. The data logger should also record security-related events.

ICE Equipment Interfaces declare the functional capabilities of the device (e.g., format of its data streams, commands to which it responds) along with non-functional properties of the data such as the rate at which data elements are streamed from the device. It is crucial that ICE Interfaces are designed with considerations for usable security, for both developers and clinical end-users.

2.2 Background on Data Distribution Service: A Communication Platform for ICE

Many communication standards have been proposed for dynamically composable and interoperable medical devices and information systems. Unfortunately, few of them include security mechanisms that are flexible and comprehensive enough to meet current and future safety needs [2]. In fact, recent work [2] has shown that there are significant gaps between required security properties for these systems and those that can be addressed even by a combination of currently standardized protocols. Safety considerations in these standardization efforts are effectively incomplete due to a lack of appropriate security analysis. Unfortunately, the promising ICE standard is no different. To address this, we developed a prototype of ICE based on RTI's implementation of the Object Management Group (OMG) Data Distribution Service (DDS) [3] as the ICE Network Controller, with the hopes of identifying & addressing a number of such gaps.

DDS is a communications API and an interoperability standard that provides a data-centric publish-subscribe model for integrating loosely coupled real-time distributed systems. A key feature of DDS is that it is *data-centric* in the sense that it separates state management and data distribution from application logic and supports discoverable data models. This exposes the data model to the communication middleware, enabling the DDS middleware to reason about and optimize the performance of data movement within the system. In order to customize run-time behavior and achieve a desired performance profile, DDS allows publishing and subscribing entities to express several quality-of-service (QoS) parameters. The offered versus requested QoS requirements of the participating entities are matched before any communication can proceed. The standard DDS QoS parameters include durability, reliability, deadline, resource limits, ownership, liveliness and several others [3].

DDS is currently being used as an Industrial Internet connectivity platform in many critical applications [10] within healthcare [5][11][12] [13][14][15][16][17][24][25], energy [21], transportation [20], and defense [22] sectors.

2.3 Data Distribution Service Security

The OMG DDS Security Specification adds support for authentication, authorization, access control, confidentiality, integrity and non-repudiation for the data sent over DDS. Moreover, it provides a security auditing capability to evaluate the overall communication state. Due to the data centric design of DDS, DDS Security can provide *fine-grained access control* over the messages and sub-messages that include both data and meta-data. This allows DDS to control and enforce which applications have authorization to publish and subscribe to the numerous data types on the network.

DDS Security is designed to handle scalable deployment scenarios, specifically the one-to-many (multicast) distribution of encrypted information while maintaining real-time quality-of-service. It also provides an extensible plugin-based architecture, as well as a set of built-in plugins for out-of-the-box interoperability. This architecture allows application developers to integrate with pre-existing identity management mechanisms, authorization policy repositories or cryptographic libraries, which might be program-specific.

Figure 2 shows the pluggable architecture of DDS Security. The authentication plugin supports identity verification, mutual authentication and shared secret establishment. The access control plugin enforces granular security policies. The cryptographic operations, such as encryption, decryption, hashing, digital signatures and key derivation are implemented in the cryptographic plugin. Finally, logging and data tagging plugins are used for auditing security-relevant events and annotating data with a security label, respectively.

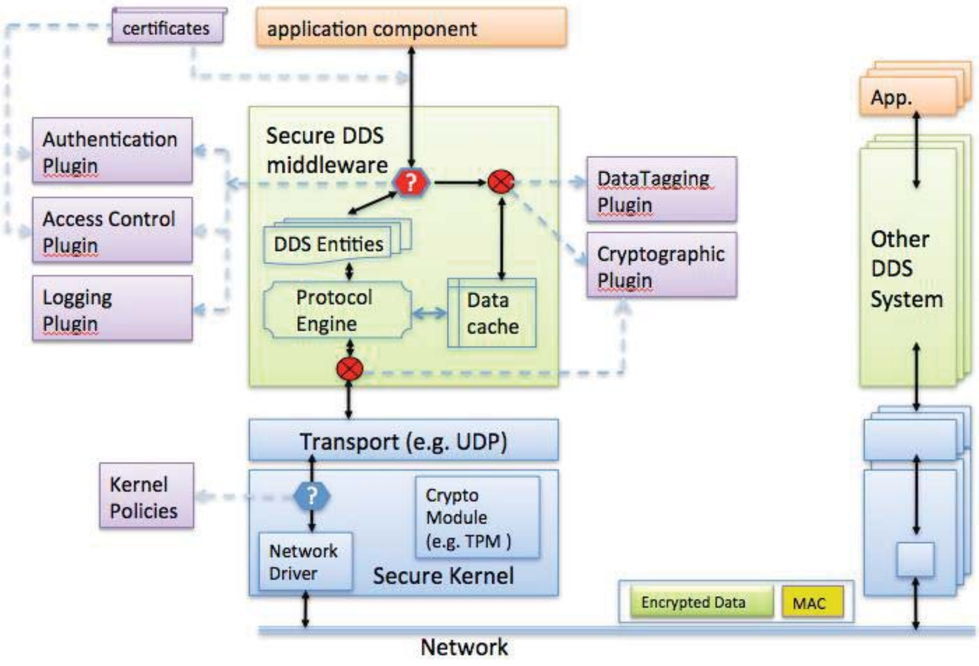


Figure 2: Architectural View of DDS Security

3. Primary Analysis

Our foremost objective towards laying down the foundation to secure clinical environments was to identify security risks, threats and requirements of various clinical scenarios. These are listed in the table below. Our findings have been mostly consistent with some of the existing literature on the topic [6][7] as far as external attackers are considered. However, we found that an important yet often neglected requirement is to minimize the impact of insider attacks posed by already-compromised devices that are unknowingly used in ICE settings. We discuss such an attack to instantiations of ICE that utilize secure transports such as TLS in Section 4.

Attack Class	Description	Susceptible Components
Destroy	Physically destroy ICE components; e.g. cut an infusion pump tube.	All Architectural Components of ICE
Disturb	Modify exchanged data to prevent correct operation of components; e.g. man-in-the-middle or replay attacks	All Architectural Components of ICE
Reprogram	Modify data or code in an ICE component to prevent its correct operation; e.g. modify infusion pump software to deliver extra medication	All Architectural Components of ICE Except the Communication Network Itself
Denial of Service	Exploit bugs or interfaces that were not designed with security in mind	All Architectural Components of ICE
Eavesdrop	Listen in on the deployed ICE environment to learn sensitive information.	Communication Network

Table 1. General attack model for ICE as identified in [6]

Use of an ICE controller based on DDS Security potentially addresses or mitigates Disturb, Denial of Service and Eavesdrop attacks. Further, it would mitigate the impact of insider attacks dramatically.

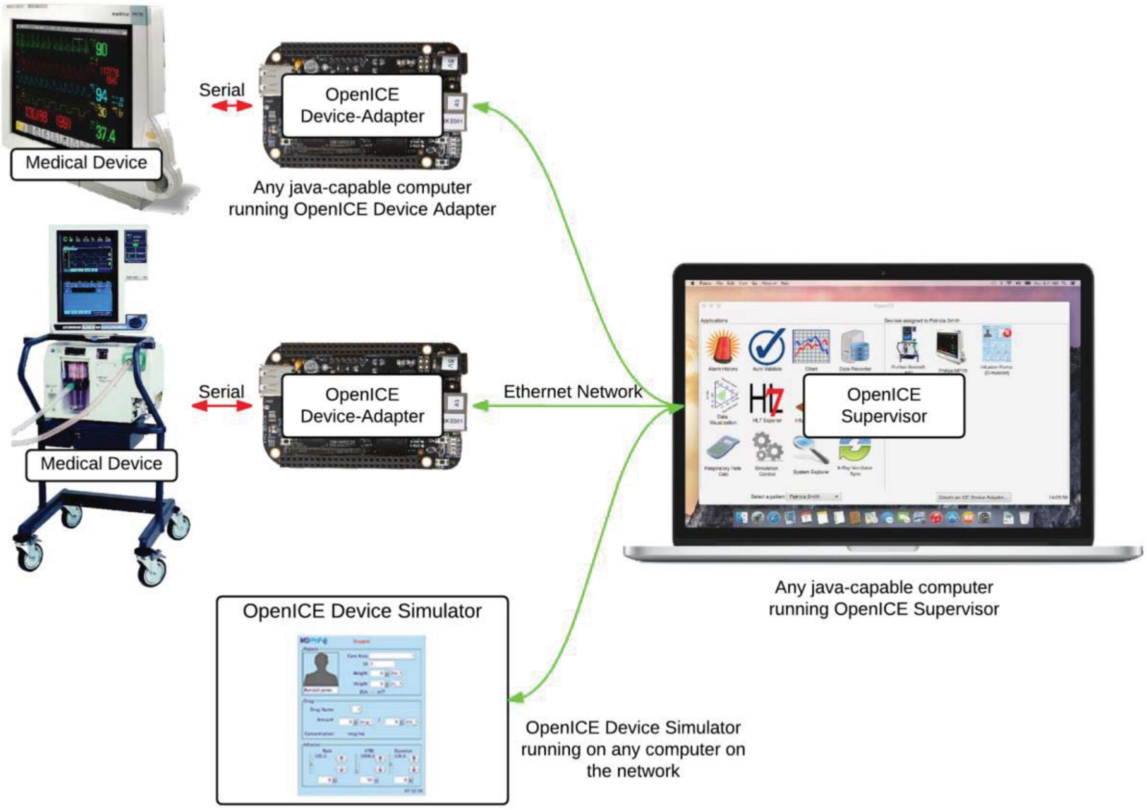

Serial
OpenICE Device-Adapter
Medical Device
Any java-capable computer running OpenICE Device Adapter
Serial
OpenICE Device-Adapter
Ethernet Network
Medical Device
OpenICE Supervisor
Any java-capable computer running OpenICE Supervisor
OpenICE Device Simulator
OpenICE Device Simulator running on any computer on the network

Figure 3. OpenICE – Developed by MD PnP Lab, it enables connectivity between various types of devices.

4. Prototype Development

We designed and implemented two different prototypes, each supporting all of the medical applications (e.g. PCA Safety & Smart Alarms) provided in the OpenICE environment. OpenICE is an open-source reference implementation of ICE released by MD PnP lab. Figure 3 shows how OpenICE enables connectivity between various types of devices.

4.1 Practical Security Attacks on Current OpenICE Platform

Prior to the development of the prototypes, we verified that OpenICE, without any explicit security measure, can be easily attacked, endangering patient safety and privacy. We developed customized sniffers and injectors that an external attacker could use to eavesdrop on ICE communications or disturb device behavior (e.g. stop drug infusion, or inject wrong sensor readings).

4.2 First Prototype: OpenICE Using DDS on Top of Secure Transports

Our first prototype integrates OpenICE with RTI Connext DDS as the Network Controller, running on top of TLS or DTLS transports. In this prototype, security measures such as confidentiality or integrity of exchanged messages are not applied at the ICE Network Controller level, but at the transport level that it uses. A fundamental research question here is whether such widely used, communication protocols provide acceptable security and performance for ICE.

While transport-level security provides typically reasonable protection against external attackers, it is not without limitations. Transport-level solutions do not provide any mechanism for granular access control. Even though these solutions protect the communication channel from external eavesdropping or packet injection, they do not provide any access control mechanism for data streams happening within the same protected link. Consequently, solutions based on them are vulnerable to insider attackers, as we demonstrate in our second prototype.

Transport-level security is also not sufficiently flexible to balance security versus performance. All messages that pass through the established secure link will be encrypted and authenticated, imposing an overhead that may not be necessary in many use cases. For example, risk analysis of an ICE system might conclude that encrypting temperature values from a sensor in a public room is not required and it is only needed to make sure sensor readings are authenticated. Being able to fine-tune security measures based on risk is especially important for resource-constrained devices or large-scale ICE or MIoT systems with bandwidth or delay sensitive applications. Further, such fine-tuning should ideally happen with minimal, if any, changes to the code base, as the code may not be available for modification or too costly to be modified.

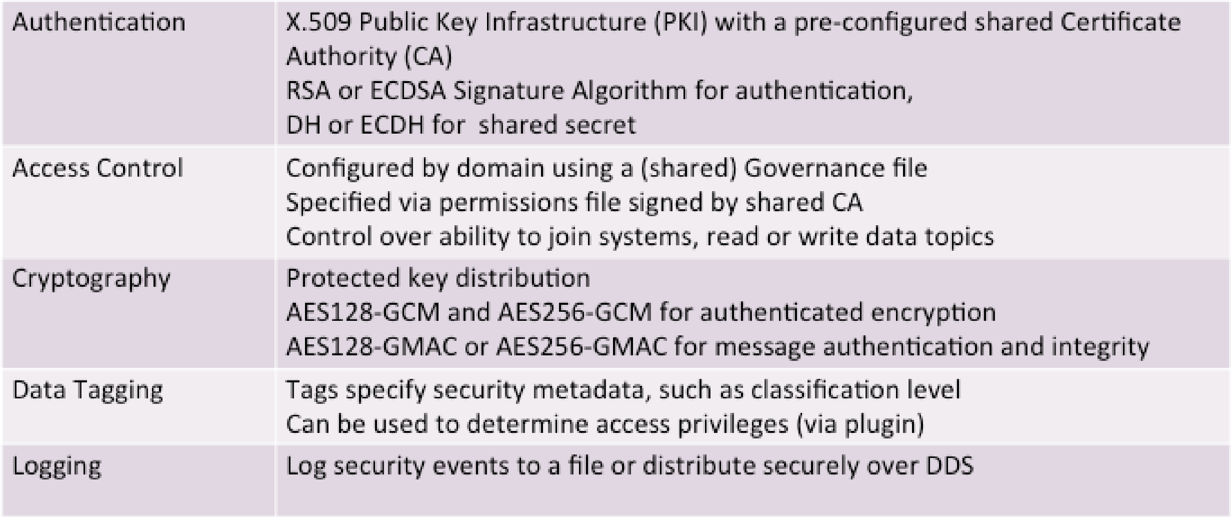
Another issue with widely used transport-level security solutions such as TLS and DTLS is the lack of support for multicast. Multicast support has proven extremely useful for efficient and scalable discovery and information exchange in industrial systems.

4.3 Second Prototype: OpenICE Using RTI Connext DDS Secure

In the second prototype, we integrated OpenICE with RTI's implementation of the beta version of DDS Security Specification as the Network Controller. We also made sure that the integrated solution works with RTI Routing Service, acting as an intelligent gateway connecting multiple ICE environments. Such integration would ease adoption of ICE in fragmented hospital networks or in cases where ICE systems belong to different administrative domains.

RTI Routing Service is a software solution that provides the ability for unmodified new and legacy applications to interoperate, even if they were not originally designed to work together. It can be used to integrate different system or bridge to legacy messaging and networking technologies. It is used to form logical partitions for DDS systems across LANs or WANs or to bridge non-DDS systems provided that appropriate DDS adapters are linked to it [10]. Utilizing the Routing Service as an intelligent gateway enables a variety of security administration use cases in ICE. An example would be to segregate insecure legacy medical devices into separate administrative domains without disconnecting them from the secure ICE environment. This allows for a different, likely more strict, set of security policies to be applied to the legacy devices, while still keeping them connected to ICE.

We used DDS Security Built-in Plugins to protect ICE Network Controller operations. Table 2 shows capabilities of built-in plugins.



Authentication	X.509 Public Key Infrastructure (PKI) with a pre-configured shared Certificate Authority (CA) RSA or ECDSA Signature Algorithm for authentication, DH or ECDH for shared secret
Access Control	Configured by domain using a (shared) Governance file Specified via permissions file signed by shared CA Control over ability to join systems, read or write data topics
Cryptography	Protected key distribution AES128-GCM and AES256-GCM for authenticated encryption AES128-GMAC or AES256-GMAC for message authentication and integrity
Data Tagging	Tags specify security metadata, such as classification level Can be used to determine access privileges (via plugin)
Logging	Log security events to a file or distribute securely over DDS

Table 2: Capabilities of DDS Security built-in Plugins

The current commercial DDS Security Plugins rely on an existing public-key infrastructure (PKI) to be in place. Management of the PKI is outside the scope of DDS Security and industry best practices can be used. For our prototypes, we used a self-signed certificate authority.

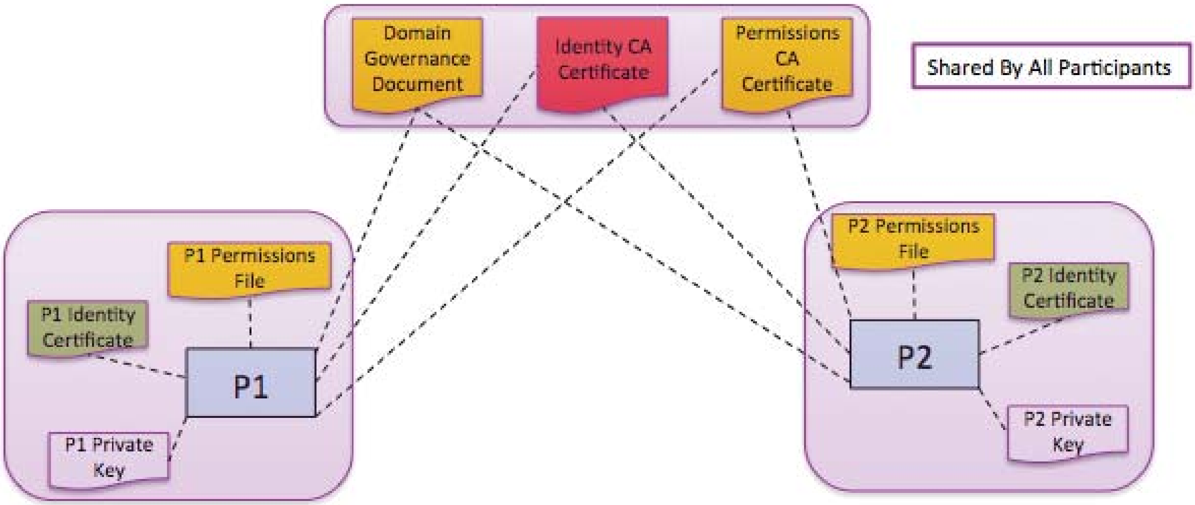


Figure 4: Deployment and configuration of DDS Security for two participants.

To operate using RTI's built-in security plugins, each DDS Domain Participant[1] requires 1) a public/private key pair, with the public key signed by a trusted certificate authority (referred to as the identity CA) forming an identity certificate, 2) permissions file signed by a trusted certificate authority (referred to as the permissions CA), 3) a DDS domain governance file, signed by the permissions CA, 4) an Identity certificate of the Identity CA, and 5) an Identity certificate of the Permissions CA. Figure 4 shows a possible deployment of DDS Security with two participants, P1 and P2. In an ICE setting, they could be an Oximeter and an ICE supervisor respectively.

The domain governance document is written in XML (eXtensible Markup Language), and specifies which DDS domains shall be protected, along with the details of the protection. The domain governance document is signed by the permissions CA and configures the following security aspects of the DDS domain: whether the discovery information should be protected and the kind of protection (MAC or ENCRYPT_THEN_MAC); whether liveliness messages should be protected; whether a discovered participant that cannot authenticate or fails authentication should be allowed to join the domain and see any data configured as unprotected; whether metadata (e.g., sequence numbers, heartbeats) should be protected and how; whether the payload should be protected and how; and whether read/write access to the topics should be open to all or restricted to the participants with proper permissions.

The XML permissions document contains the permissions of the domain participant, including which DDS domains it can join, what topics it can read or write, and what tags are associated with it.

4.4 Security Attacks on ICE When Run on Secure Transports

We implemented effective attacks against ICE when the Network Controller uses secure transports such as TLS or DTLS. Our attacks are based on ICE Infusion Safety App, which utilizes closed loop control of medical devices for safe delivery of patient controlled analgesia (PCA). The application controls the administration of IV medication and is programmed to stop the pump if it detects that the patient is in a non-normal state. Patient state is inferred from the readings of devices such as oximeters and capnographs. Figure 7 demonstrates a simplified ICE infusion safety application scenario, with topics published to or subscribed by various ICE components (see OpenICE Infusion Safety App Architecture [26] for further details).

[1] A DDS *domain* is a concept used to bind individual applications together for communication. To communicate with each other, *DataWriters* and *DataReaders* must have the same *Topic* of the same data type and be members of the same *domain*. Applications in one domain cannot subscribe to data published in a different domain. DomainParticipant objects enable an application to exchange messages within domains. *DomainParticipants* are used to create and use *Topics, Publishers, DataWriters, Subscribers,* and *DataReaders* in the corresponding *domain*.

In our attack, a compromised pulse oximeter publishes Alarm Limits associated with an uncompromised capnograph, either masking an alarm when it should happen (e.g. in case of a drug overdose) or when it shouldn't (e.g. causing alarm fatigue). Even though all communication in this attack scenario is encrypted and authenticated, a compromised insider device can cause system-wide damage, simply because what it can or cannot publish is not enforceable. DDS Security allows for fine-grained access control per device, preventing this significant type of attack.

In the second prototype, each ICE device has a cryptographically signed permission file that specifically indicates what topics can be published or subscribed by it. In order to recreate the original attack on this new framework, the attacker would have to hack into the public-key infrastructure (PKI) used in the framework, which is considered a much more difficult task if PKI is managed properly. In any case, if the PKI infrastructure becomes compromised, any cryptographic approach based on it will fail, be it based on TLS/DTLS or DDS Security.

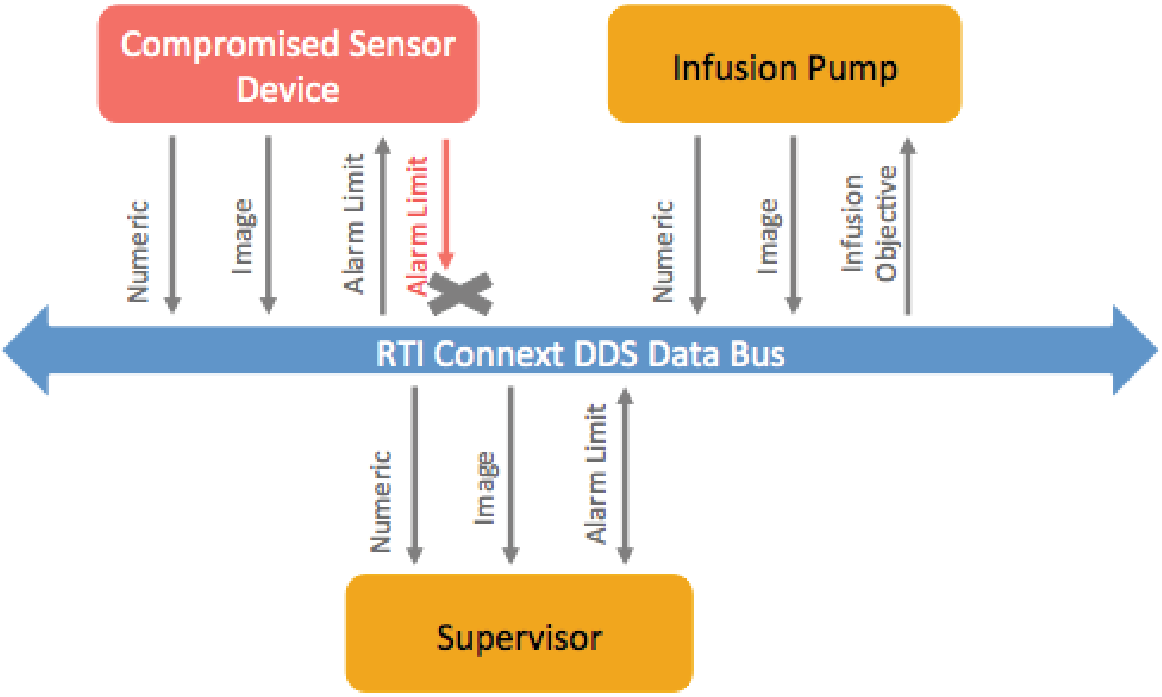


Figure 7. Simplified Architectural Diagram of OpenICE Infusion Safety App

Boxes represent ICE devices, and arrows represent topics that each device either publishes or subscribes to. The box in red represents a compromised oximeter that, in principle, should not be allowed to publish AlarmLimit topic data. AlarmLimit topic data should only be published by the ICE supervisor and no other device, even if they are correctly authenticated. Both DDS Security and a secure transport such as TLS/DTLS allow for certificate-based authentication of devices, but use of DDS Security also enforces granular access control. Granular access control provides further resilience in presence of insider attackers, preventing system-wide damage such as the one discussed above.

5. Conclusion & Future Work

The grand vision of the Medical Internet of Things is to enable the deployment of patient-centric and context-aware networked medical systems in all care environments, ranging from homes and general hospital floors, to operating rooms and intensive care units. The key to realizing this vision is to come up with standardized architectures that balance utility, reliability and safety requirements with those of security and privacy. The ICE framework, as defined by the ASTM F2761-09 standard is definitely an important step toward enabling interoperable MIoT, however, it does not yet explicitly address security concerns.

In this paper, we presented recent research on protecting communications within IICE based on the fine-grained security mechanisms provided by the OMG DDS Security specification. We developed the two prototypes that respectively utilize secure transports (TLS/DTLS) and the DDS Security Architecture, and demonstrated why transport-level security solutions may not provide sufficient resilience against insider attacks utilizing authenticated but compromised medical devices.

In the future, we will work on defining and enforcing holistic security policies for ICE, integrate with endpoint protection mechanisms (e.g. secure Operating Systems, hardware-based root of trust), integrate with security management and monitoring solutions and explore issues at the intersection of usability and security in MIoT systems in general and ICE systems in particular.

6. References

[1] ASTM F2761, 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, 2013.

[2] Foo Kune, D. a. (2012). Toward a Safe Integrated Clinical Environment: A Communication Security Perspective. Proceedings of the 2012 ACM Workshop on Medical Communication Systems (pp. 7--12). New York: ACM.

[3] OMG Data Distribution Service Standard: http://www.omg.org/spec/DDS/1.2/

[4] OpenICE: https://www.openice.info/

[5] RTI Customer Snapshot: DocBox: http://www.rti.com/docs/DocBox.pdf

[6] K. K. Venkatasubramanian et al. "Security and Interoperable-Medical- Device Systems, Part 1," IEEE Security Privacy, vol. 10, no. 5, pp. 61-63, Sept./Oct. 2012.

[7] Eugene Y. Vasserman , Krishna K. Venkatasubramanian , Oleg Sokolsky , Insup Lee, Security and Interoperable-Medical-Device Systems, Part 2: Failures, Consequences, and Classification, IEEE Security and Privacy, v.10 n.6, p.70-73, November 2012.

[8] Content of Premarket Submissions for Management of Cybersecurity in Medical Devices, Guidance for Industry & FDA Staff, October 2014

[9] Postmarket Management of Cybersecurity in Medical Devices, Draft Guidance for Industry and FDA Staff, January 2016

[10] RTI Customers: https://www.rti.com/industries/index.html

[11] ROS on DDS http://design.ros2.org/articles/ros_on_dds.html

[12] RTI Customer Snapshot: EMS Device Integration Platform for World's largest EMS equipment Provider. https://www.rti.com/industries/#HEALTH

[13] RTI Customer Snapshot: Minimally Invasive Robotic Surgery. http://www.rti.com/docs/German_Aerospace_Center_DLR.pdf

[14] RTI Customer Snapshot: Exelis C4i Command and Control Systems. https://www.rti.com/industries/#HEALTH

[15] RTI Press release: GE Healthcare. https://www.rti.com/company/news/ge-healthcare.html

[16] RTI Customer Snapshot: Medical Imaging. https://www.rti.com/industries/#HEALTH

[17] RTI Customer Snapshot: Advanced Proton Therapy. https://www.rti.com/industries/#HEALTH

[18] James, John T. PhD. A New, Evidence-based Estimate of Patient Harms Associated with Hospital Care. Journal of Patient Safety, September 2013. http://journals.lww.com/journalpatientsafety/Fulltext/2013/09000/A_New,_Evidence_based_Estimate_of_Patient_Harms.2.aspx

[19] Healthcare Technology. "Deaths by medical mistakes hit records" http://www.healthcareitnews.com/news/deaths-by-medical-mistakes-hit-records

[20] RTI IIoT Transportation Applications:. https://www.rti.com/industries/#TRANSPORT

[21] RTI IIoT Energy Applications: https://www.rti.com/industries/#ENERGY

[22] RTI IIoT Defense Applications: https://www.rti.com/industries/#DEFENSE

[23] Defense in Depth: A practical strategy for achieving Information Assurance in today's highly networked environments" https://www.nsa.gov/ia/_files/support/defenseindepth.pdf

[24] SmartAmerica Closed-Loop Healthcare Group: http://www.mdpnp.org/smartamerica.php

[25] OpenICE Infusion Safety App Architecture: https://www.openice.info/docs/3_apps.html - infusion-safety

[26] Mullen, A. B. (2013, 09). Premature enforcement of CDRH's draft cybersecurity guidance. http://www.fdalawblog.net/fda_law_blog_hyman_phelps/2013/09/premature-enforcement-of-cdrhs-draft-cybersecurity-guidance.html

[27] M. Rushanan, D. F. Kune, C. M. Swanson, and A. D. Rubin, "SoK: Security and Privacy in Implantable Medical Devices and Body Area Networks", IEEE Symposium on Security and Privacy, 2014

IEEE Journal of Translational Engineering in Health and Medicine

Received 2 February 2016; revised 6 July 2016; accepted 8 July 2016. Date of publication 8 August 2016;
date of current version 24 August 2016.

Digital Object Identifier 10.1109/JTEHM.2016.2596283

The Importance of State and Context in Safe Interoperable Medical Systems

SANDY WEININGER[1], (Senior Member, IEEE), MICHAEL B. JAFFE[2], (Senior Member, IEEE), MICHAEL ROBKIN[3], TRACY RAUSCH[4], (Member, IEEE), DAVID ARNEY[2], (Member, IEEE), AND JULIAN M. GOLDMAN[2], (Member, IEEE)

[1]Office of Science and Engineering Laboratories, FDA/CDRH, Silver Spring, MD 20993, USA
[2]MDPnP Program, Department of Anesthesia, Massachusetts General Hospital, Boston, MA 02114, USA
[3]Anakena Solutions, Inc., Woodland Hills, CA 91367, USA
[4]DocBox, Inc., Newton, MA 02460, USA

CORRESPONDING AUTHOR: S. WEININGER (sandy.weininger@fda.hhs.gov)

This work was supported in part by NIH under 5U01EB012470-03, in part by NSF under IIS-1239242, and DOD under W81XWH-09-1-0705, W81XWH-12-C-0154, W81XWH-11C-077, and W81XWH-13C-0107.

ABSTRACT This paper describes why "device state" and "patient context" information are necessary components of device models for safe interoperability. This paper includes a discussion of the importance of describing the roles of devices with respect to interactions (including human user workflows involving devices, and device to device communication) within a system, particularly those intended for use at the point-of-care, and how this role information is communicated. In addition, it describes the importance of clinical scenarios in creating device models for interoperable devices.

INDEX TERMS Safety, interoperability, state, context.

I. INTRODUCTION

It is useful to apply a model of interoperability, such as the Levels of Conceptual Interoperability Model (LCIM) [1] to better conceptualize the challenges of medical device interoperability, and help understand the information needed to achieve safety in the medical application space. Tolk's model, consisting of five levels, was later extended to seven levels by Turnitsa [2] to characterize the types of interactions taking place both within the system and externally. Tolk and Turnitsa postulated that there are operations that need to be performed to enable safe interoperability, regardless of whether these operations were performed by human designers, human operators or interacting devices. If a device is designed to act in the place of the human, it is logical the device model would need to include at least the same information that the human considered to assure the safety of the patient. This information relates to the state of the devices and patient, the context of clinical care, and all relevant assumptions relating to the associated hazards, risks and mitigations.

Robkin *et al.* [3] applied Tolk's and Turnitsa's model of conceptual interoperability to medical devices and healthcare, and further showed that for systems to achieve an appropriate level of interoperability, the design process of the developers (of the interoperable components and interfaces) must be working at a higher level of interoperability. Current healthcare delivery systems are replete with examples of both successful and unsuccessful interoperability. Some of these examples are a failure of semantic interoperability, in which data is interpreted differently by the two parties, or of dynamic interoperability, where the two organizations don't have a shared understanding of the clinical state, and context. We postulate that the exchange of information via a robust device model requires dynamic interoperability and is necessary to achieve safe interoperability. This paper examines the use of clinical scenarios to capture, characterize, and make this information explicitly available through an electronic data interface (EDI).

In "Solving the Interoperability Challenge" [4], Goldman highlights the challenges of interoperability,[1] the ability of medical devices, clinical systems, or their components to communicate in order to safely fulfill an intended purpose [5], which is illustrated by the slow pace the medical device industry has taken to achieve plug-and-play (PnP) (i.e. seamless device interfacing based solely on configuration, not custom software development). Goldman proposes an approach in which the clinical community defines interoperable

[1] An alternative defined per IEC: Capability of objects to collaborate, that is, the capability mutually to communicate information in order to exchange events, proposals, requests, results, commitments and flows (per ISO/IEC 10746-2: 2009).

clinical scenarios where the workflow and interactions, both between the human operators and devices and between devices are appropriately specified and hazardous situations that may arise are identified. The shared use and understanding of clinical scenarios helps manufacturers, researchers, standards development organizations to achieve conceptual interoperability between themselves - that is they will have identical understanding of their mutual goals, scope, and constraints, and shared understanding of the relevant patient context. The implementation of these clinically meaningful scenarios, which are enabled by interoperable components, can be used as a metric for the adoption of interoperability. A recent AAMI publication [5] in its overview of the landscape of medical device interoperability including standards, stakeholders and issues, notes the major challenge in achieving medical device interoperability is enabling medical devices to communicate in a meaningful way. A key component of enabling meaningful device communication is the creation and use of device models that include device attributes and clinical context.

II. MEDICAL DEVICE INTEROPERABILITY EXAMPLES

Two examples of interoperable components in current use in the medical device industry will help illustrate the benefits, challenges and importance of achieving widespread adoption of interoperability. The standardized use of the R-wave of the ECG for synchronizing the delivery of defibrillation pulses to the patient (see [6]-section 104) or the de-facto standard for the fetal ECG digital interface (with physical/data link and application layers specified) [7] are examples of the successful application of interoperability to enable a clinical application. The safety of these de-facto standards, specifically in terms of the plug-and-play interoperability (as opposed to the safety of the medical device or the connection itself), may not have been considered during the design of the components, but were accepted over time via demonstrated safe interoperability proven through widespread use, leading to adoption by numerous third parties. This could not have been achieved if the interface specifications had not been available to third party device manufacturers, and the context, intent, and use of the connectors well known to both device and the connector manufacturers. In the absence of the exchange of formal specifications, all parties must have the same understanding of context, intent, and use to prevent failures and unintended interactions.

III. CHALLENGES IN WIDER DEPLOYMENT OF MEDICAL DEVICE INTEROPERABILITY

The ASTM standard F2761, “Medical Devices and Medical Systems — Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) ([8], Annex B) describes several clinical scenarios that lack interoperability and lead to unsafe clinical outcomes. The application of concepts used in the distributed processing field [9] can help to analyze these clinical scenarios. These concepts require knowledge of design aspects of the medical device and its associated capabilities, which F2761 refers to as a device model. It asserts that the F2761 device model includes sufficient information for communicating state and context information to enable the safe clinical use of the system comprising components, each with their associated device models. It is important to realize that the system device model (e.g. ICE device model) may have more information (e.g. physical location of device sensors and body position, care location such as ICU or home environment) than the sum of the component device models. Integrated systems can enable monitoring of device behaviors and clinical effects of nefarious activity, such as resulting from cybersecurity vulnerabilities. In this perspective, comprehensive, contextually rich data from networked devices, enables improved cybersecurity.

Platforms supporting plug and play (PnP) allow devices and applications to be connected or disconnected on-demand by supplying the underlying physical and data infrastructure to communicate information and control signals between devices and applications. PnP systems allow for the development of “apps” that reuse existing infrastructure. Goldman’s goal of creating “Lego blocks” (http://psqh.com/janfeb08/connected.html) for developing safe and interoperable healthcare applications is dependent in part on the ability of the device model to adequately represent the information required for safe use. The integrated clinical environment (ICE) standard [8] describes a system architecture and specifies the need and overall structure of a device model to represent the capabilities of the equipment and serve as the repository for this essential information including information to qualitatively and quantitatively describe, control and monitor the system. Reasons that PnP is not widely implemented in the medical device space is due in part to the absence of a formal, widely adopted definition of a device model, and an ecosystem for development and maintenance. These two pieces of the puzzle need to be resolved and clearly defined to achieve safe interoperability through PnP. The use of device models to enable reliable communication and data sharing between components within a system are well known and previously used in: (a) industrial bus standards [10] to allow data sharing between devices (e.g. controllers, data acquisition components, sensors) of different types and generations and (b) standards supporting the sharing of electronic health information (e.g. HL7 RIM) [11] based on a static reference model of health and healthcare information.

Device models are only one means of encapsulating the information needed to help assure reliable data sharing and, with respect to medical devices, helping assure safe interoperability. They provide the system designers content and context on how a component should be used, describe the operation of the device relative to the interactions that occur in the clinical scenario, and document the information that must be conveyed or received. When device models are shared across a community, they enable platforms and their components to be leveraged and reused. Device model re-use may lead to more rapid and efficient product development. An important caveat about the use of device models is that all parties (e.g. components) involved in an interaction has to possess the same understanding of the device model or unexpected and unintended consequences can emerge.

IV. DETERMINING RELEVANT ATTRIBUTES OF A DEVICE MODEL RELATED TO SAFETY

A medical device may simultaneously function as a standalone device and as a component in a larger system. Therefore, it is important that this device is able to communicate the content of its device model via the EDI. The device model of a component, from a systems perspective, contains functional and non-functional specifications governing the safe operation of that component in the system. In addition to the information identifying the device hardware, software and connected accessories, various aspects of the ''internal'' operations of a device may be necessary to include as part of the device model. This information is identified by understanding the role the device plays in the system, the patient care environment, and what actions the human operators perform in order to maintain the safe operation of the device. The nature of the information required for a more complete and complex device model is documented in the form of clinical scenarios.

To characterize thoroughly the interactions of an interoperating device, a range of clinical scenarios should be investigated involving multiple medical disciplines, clinical environments, and including hazard information. The interactions between device and system components studied could include considerations such as: the use of legacy devices, plug and play capabilities, the ability to transfer device settings from one clinical venue to another, (physiologic) closed loop control, safety interlocks, the ability of a system to capture adverse event information, time synchronization, alarms (parameter, types), decision support, and either workflow or checklist enforcement. The clinical environments surveyed should be sufficiently broad to include the intended uses of the device or system such as the operating room, intensive care unit, post anesthesia care unit, outpatient, emergency department, transport, and home care.

Given the distributed nature of platform-based systems such as those anticipated by ASTM F2761, it is appropriate to apply the terminology and approaches outlined in the existing body of knowledge for open distributed processing of information technology to safe medical device interoperability. The standard ISO/IEC 10746-2-2009, RM-ODP Reference Model Open Distributed Processing defines a model in which the objects in the system identified from the clinical scenarios, the interfaces, their interaction points, and the behaviors of each object (a behavior is a collection of actions together contained with a set of constraints) are determined. To achieve safe plug-and-play dynamic interoperability, device and patient states, and use context must also be gathered and exchanged [3].

A contract has been described as ''*An agreement governing part of the collective behavior of a set of objects. A contract specifies obligations, permissions and prohibitions for the objects involved*.'' (ISO/IEC 10746-2 clause 13.2.1) [9]. This standard specifies two types of obligations or contractual behaviors: implicit and explicit. The current behaviors and proposed future states described in clinical scenarios may be classified based upon relationships amongst the components of an interoperable system: those relationships that assume an explicit knowledge (''resulting from actions as defined in the contract'') of the contract and those that rely on an implicit contract (transparent to at least one party). The exchange of information in a contractual manner, either explicitly during the operation of the system or implicitly with all parties having the same understanding of the model, allows higher levels of safety to be achieved. For this to occur, it is necessary a contract established during system design governs the behavior of the components. The device model is one means to represent the content of the contract. It should have sufficient information about state and context to support the safe operation of the system.

V. EXAMPLES OF SHARED DEVICE MODELS

Two parties (e.g. devices or system components) may have a contract (e.g. ''use the same device model'') and still the overall system can be considered unsafe. For example, a sensor or algorithm problem may result in spurious data generated and transmitted by a pulse oximeter, despite the fact that both parties, the sender and the receiver, are using the same data and agree on the syntax and semantics of that data. To achieve higher levels of interoperability and safety, the device model may need to include additional information such as signal quality, data constraints (e.g. parameter limits), event rules (e.g. alarms thresholds), states of devices (e.g. amplification or filter modes), pre-post conditions (e.g. remote or adaptive updates to thresholds), and temporal properties (e.g. ordering of commands, events). Interactions that rely on an implicit contract-one that is assumed to be in-place rather than verified-may be, in certain cases, acceptable; but only for systems that tolerate mismatches between the interactions of system components. In order to better assure safe and successful device interactions when changing the clinical context, or device state, explicitly shared knowledge of a device model may be needed. This distinction is presented and considered necessary because devices from different manufacturers may use different device models and may not have been designed with a device model in mind (i.e. not manufactured with the intent to be interoperable). The transmission of data from legacy devices have at times been problematic because the semantics, context, and state of the legacy device may be assumed; however, assumptions will lead to incorrect data, missed data, or misinterpretation (e.g. time synchronization, assuming there are no oxygen desaturations when in fact the filter was set to maximum smoothing thereby hiding transient desaturations).

VI. INTERACTION PATTERNS USING DEVICE MODELS TO SUPPORT PNP INTEROPERABILITY

A series of interaction patterns are presented below illustrating the implicit versus explicit nature of the contract between ''objects'' and the important role that state and context, with both having static or dynamic aspects, play in defining the attributes of the underlying device model supporting safety. This illustrates that an essential component of the device model is to capture the behaviors (actions) taken by the system components as they interact. This interoperability behavior (IB) consists of two sender/receiver component interactions and the simplest type of (i.e. no behavior) component interactions. The two primary component interaction types are:

1) (IB 1) the sender and receiver are using an implicit device model (i.e. the sender is broadcasting to anyone who will listen; the receiver is listening to anyone who is broadcasting). In this IB, there is no attempt by the sender to change the state of the receiver.

2) (IB 2) the sender has an explicit contract with the receiver and may intend to change the state of the receiver.

The simplest cases are where a device does nothing (doesn't send or receive) or simply passes information from another source. A stand-alone unconnected device exhibits the simplest behavior. A human needs to perform any desired interactions between the devices. Many current devices exhibit behaviors per IB 1, transmitting data to a repository according to its own device model without regard to the capability of the connected receiving device to interpret that data. This behavior sends data with no intention of changing a receiver's state (Figure 1). For example, devices often send data to an electronic health record, where the performance and semantics of the physiologic signals are assumed to be known by all who use the system.

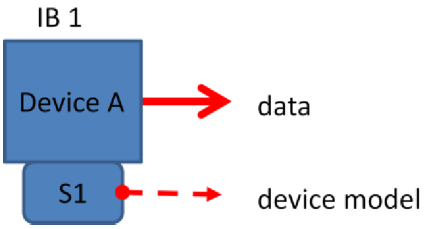


FIGURE 1. **Interoperability Behavior (IB) 1 – Device A broadcasts data according to device model S1 using implicitly assumed semantics and is at least syntactically interoperable with the receivers.**

The second behavior, IB 2 (Figure 2), involves receiving data from a broadcaster. The receiving device does not know anything about the sender's device model or if the data is valid, only that if data has appeared on its EDI. For example, some legacy medical devices may use RS232 without flow control to transmit from their functional connection. Any receiving system may not comprehend the relevance or reliability of this transmitted data.

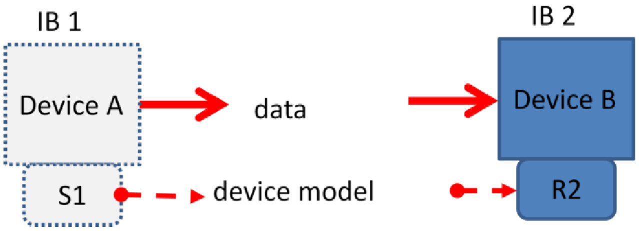


FIGURE 2. **IB 2 – Device B receives data and interprets it as device model R2 and is unaware of the sender's state or device model S1 (implied relationship-ISO/IEC 10746-2) Device B is at least syntactically interoperable with the senders device model S1.**

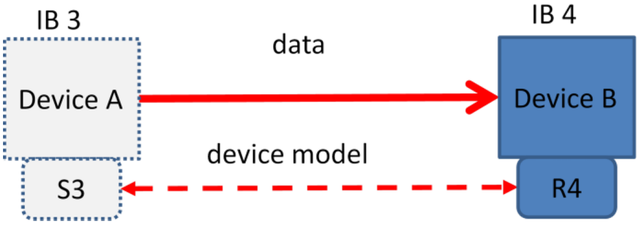


FIGURE 3. **IB 3/4 – Device A sends data (IB 3) with a device model S3 with the intention of changing Device B's state. Device B receives the information (IB 4) and acts on its content changing its own state; device model R4 is dynamically interoperable with device model S3.**

The third behavior, IB 3 (Figure 3), involves a sending device that intends to change the receiver's state and therefore must have an explicit contract with the receiver.

IB 4 (Figure 3) is the receiving part of the exchange-it changes its state based on the incoming data and likewise must have a contract with the sender. As an example, a closed loop oxygen controller in the system, to which the pulse oximeter/controller sends a command to change the delivered oxygen concentration, is an example of this interaction. An IB 4 receiver can accept information from an IB 1 sender but will have to perform extra work to assess its validity (perhaps by getting confirmation from the operator that the information is valid).

IB3/4 adds the need for a contract between the sender and the receiver to agree upon what the device model is. IB's 1-2 require syntactic interoperability while IB's 3-4 require dynamic interoperability (a shared understanding of each other's states and context).

VII. DEVICE MODEL CONTENT-CATEGORIES OF INFORMATION

It is up to the manufacturer of the application who is asserting a safety or performance claim to determine the suitability of components for use in their system and specify acceptable device models. The information to be included in the device model is wide ranging and can include [8]:

- Physiologic parameters and waveforms
- Technical and physiologic alarm status and conditions
- Device control functions
- Patient information
- Technical information
    - o Real-time/quality of service information
    - o Security controls
    - o Data constraints
- Clinical context
- Device state
- Role of the device (in terms of sender/receiver)

The categories of data that may be defined for and available on request from a particular device can include:

a. Parameters and units of measurement: Parameters and units of measurement used within the device (e.g. measured and derived parameters, waveform values and derived indicators)

b. Equipment identification: Information identifying the device (e.g. manufacturer, model, serial number, software and/or firmware versions)

c. Equipment Configuration: Information identifying the particular configuration of the device. (e.g. type of accessory connected)

d. Equipment specifications.

e. Equipment settings: Settings relating to the control and operation of the device.

f. Service monitoring: Indicators relating to preventative or corrective maintenance of the device and its accessories.

TABLE 1. Selected information in an exemplary pulse oximeter device model.

Category	Examples
Parameters and units of measurement	SpO_2 Pulse rate, Pulse Plethysmographic Waveform, Signal Quality Metric
Equipment identification	Manufacturer, model, serial number, software version and firmware version, unique device identifier (UDI), operating system version, anti-virus software version
Equipment configuration	Sensor Type Connected (reusable/single patient use; adult/pediatric; finger/ear), Sensor Model Connected
Equipment specifications	SpO_2 accuracy, declared ranges of SpO_2, Accuracy under motion and low perfusion, pulse rate accuracy, declared ranges of pulse rate
Equipment settings	Data Update Period, Averaging Time, Gain
Service monitoring	Remaining sensor life; next periodic maintenance date, time that real-time clock was last set

VIII. DEVICE MODEL IN REAL LIFE

In addition to these categories of interactions and data, manufacturers are encouraged to leverage the additional capabilities afforded by an external connection allowing data sharing and device interactions. These capabilities include the use of intelligent algorithms (e.g. closed loop control) that may reside in the device which may adjust internal algorithms or display settings based upon information received externally. Additionally, devices can have a dual nature-as stand-alone or as a component in a larger system by supplying data or therapy as a service. Standards have begun to take this approach further by considering broader clinical use cases and encouraging the availability of contextual information through the external interface. The most recent drafts of the continuous positive airway pressure delivery devices (ISO 80601-2-71:2015) and Pulse Oximeter (ISO/CD 80601-2-61) particular standards are two examples of this new direction. In the example of the pulse oximeter, this may include: location, status, and data from other sensors. Erroneous pulse oximeter values could be prevented if the pulse oximeter was aware that its sensor was on the same limb as the blood pressure cuff and of the cuff's real-time inflation status. An example of the content of these categories of data for this particular device, the pulse oximeter, are in Table 1.

One effort to capture the information necessary for a specific device's "device model," driven by the MD PnP research program, is the Medical Device Interface Data Sheet (MDIDS) project. The MDIDS intend to serve as an extensible reference for standards development organizations (SDOs), manufacturers, researchers, and clinical organizations. MDIDS include both "generic" and device-type sheets with all sheets including device identification and a description of the data encoding used for device-specific data elements, as well as aspects of clinical and system context. [12] (http://mdpnp.org/mdids.html) Preliminary MDIDS documents developed for devices include the pulse oximeter, critical care ventilator, anesthesia workstation, defibrillator, and dialysis machine. The data items for the MDIDS derive from existing devices capabilities and analysis of the future device capabilities necessary to support identified clinical scenarios. One device's MDIDS, "the Anesthesia Workstation" includes, in addition to the generic list, an additional 19 measurement variables and 113 alarms (www.mdpnp.org). One standardized device model construct central to the ISO/IEEE communication standard is the domain information model, based on a device-centric paradigm, which comprises objects containing the representations of the data and their relationships [13].

It is vital to characterize device behavior, especially in regard to electronic data inputs, outputs and responses to commands. This is the information captured in the MDIDS. With increasingly complex interoperable environments, it is likely that not every potentially hazardous situation will be anticipated during the design, testing, and implementation of an interoperable medical system. It is important to minimize unintended behaviors, including those deriving from emergent properties. The MDIDS information could be used by manufacturers (including app developers) to identify information available for interoperable functions, as well as for performing risk analysis. A system data logger, a key component in an ICE compliant system [8], can be used to capture these undesirable behaviors for possible inclusion in the MDIDS.

IX. SELECTED EXAMPLES ILLUSTRATING THE IMPORTANCE OF A CONTRACT BETWEEN COMPONENTS AND ITS CONTENT

This section describes five clinically relevant examples, primarily at the point-of-care, that highlight the importance of a complete contract and the necessity of a device model based on this contract to achieve a safe system (or to achieve safe interactions between devices). For each example, the information or interaction lacking is described in both a clinical context in the text and in a risk- based context in Table 2.

TABLE 2. Selected clinical scenarios.

	Knowledge of device settings	**Physiological feedback**	**Location awareness**	**Knowledge of parameter signal processing**	**Device synchronization**
Devices →	**#1)Pulse oximeter used for sleep screening**	**#2)PCA pump***	**#3)Finger pulse oximeter and BP cuff on the same arm**	**#4)EHR and medical device data**	**#5)X-Ray/Bypass and ventilation, or Oxygen and Laser devices in use***
Cause	Averaging time in a pulse oximeter is variable and may not be made known to the user/app	Architecture or device defect; device not capable of acknowledging, no feedback signal was architected into the platform	Inflation of cuff and monitoring of SpO_2 at the same time on same limb. State of NIBP not made known to App so that it can ignore the SpO_2 data	Clinically relevant events may not be captured in EHR	Failure to ventilate after X-ray procedure or resumption of cardiopulmonary bypass or failure to lower oxygen fraction during airway laser use.
App or System failure mode	Transient desaturation will be missed; data does not map/match to actual physiological parameter	Unknown whether command was followed or seen by pump. Unknown what state pump is in.	State of NIBP not known to App. It can ignore SpO_2 data. App misinterprets SpO_2 signal.	State and context may not be known	State of ventilator not known to X-ray or bypass machine. State of O_2 delivery unknown to laser device.
Local failure effect	App underdiagnoses severity of respiratory depression due to sleep apnea.	Medication overdose; App keeps seeing physiological information that is in conflict with pump stopped.	False positive desaturation leading to stopping infusion inappropriately; false positive indication of probe unreliability	Rapidly changing clinical event may not be captured in patient record leading to failure to properly treat condition	Failure to ventilate leading to hypoxia Failure to lower oxygen levels leading to airway fires
Larger systemic effect	Injury or death	Injury or death	App fails to run as intended	Injury or death	Injury or death
System hazard/ requirement	App needs to know averaging time	pump state needs to be available to confirm stop command	NIBP should communicate the status of cuff inflation and location	State and context need to be captured by EHR – including clinical data-sufficient time resolution	Ventilation Disable should be limited in time, linked to alarms and synched to therapeutic devices
General class of hazard	State of the device was not conveyed to the app	Indication of state of the device not conveyed to the app; defect in command and control.	Context (device placement) and State (measurement event synchronization) of device needed to make correct decisions	Context and state not conveyed to EHR	Operational state of devices need to be shared on a timely basis
Patient Context	Patient in sleep lab.	Hospitalized patient in bed on PCA infusion pump.	BP cuff proximal and ipsilateral to SpO_2 probe	Clinically relevant events not recorded-care may be impacted.	Ventilation Paused means patient is at risk for hypoxia. O_2 concentration not lowered means risk of fire and burns
Device and Patient States **(future)**	Device: Oximeter in “fast mode” and patient not moving enables system to distinguish oxygen desaturation from noise caused by patient movement	Device: PCA pump and physiologic sensors interoperable Patient: early respiratory depression detectable	Device: Inflated BP cuff with SpO_2 probe distally located Patient: artifact on SpO_2 identified	Device: medical device with settings known to EHR Patient: transient events recorded	Device: Ventilator with X-ray or O_2 device with Laser Patient: lack of ventilation or excess O_2 detected
Device-device interaction	Device configuration (operational state) based on clinical status	Pump must be able to process external stop command and change its own state	SpO_2 must be aware of NIBP state to avoid errors	Lack of complete data set inhibits clinical interpretation	State of O_2 device must be known, exchanged, and synchronized with laser device

* These scenarios are based on content from Annex B of ASTM F2761 [8].

Example 1: A pulse oximeter used for sleep screening. Device model is inadequate; lacks information about device model “state” (e.g. configuration settings).

Consider a clinical scenario where a pulse oximeter is supplying saturation values to another device (e.g. a decision support component) that is computing a derived index (e.g. lowest SpO_2, apnea/hypopnea index, number of respective events per hour) to assess the severity of a patient’s sleep apnea. The effectiveness of these indices, such those detecting the nadir of transient desaturations, has been shown to be dependent on the SpO_2 averaging time [14]. Therefore, the component computing the index needs to know the pulse oximeter SpO_2 averaging time so it can determine whether it is appropriately set for the current patient state. If the device sending the information and the component receiving the information and computing the index are developed without a common understanding of those device states and the patient context (IB 1 and 2 respectively, i.e. they don't share the device model for the specific pulse oximeter in use) then the performance of the component computing the index may not meet the sleep-analysis manufacturer's specifications. The computing component does not determine whether the data is right, it computes using the data it receives (IB 2). It is the responsibility of the system to insure that the computing component is sent the appropriate additional parameters to ensure that the correct transmitted data is based on clinically appropriate settings. In the situation with inappropriate settings, possibly determined by querying the pulse oximeter, the operator should be alerted to set the correct averaging time on the pulse oximeter (i.e. IB 1) or if the functionality is available, the computing component can configure the settings in the oximeter, illustrating IB 3.

Example 2: Patient controlled analgesia (PCA) pump interlock. This device model does not contain operational state of device or does not make state of device externally visible.

Consider a clinical scenario that implements a safety interlock to halt a PCA pump when respiratory depression is detected. In the case of an open medical platform, a software application or an ''App'' diagnoses early respiratory depression and sends a command to the pump to stop infusing prior to patient harm. If the pump interface is incompatible with the App sending the stop command, IB 1, where the pump cannot be remotely controlled, the pump cannot be stopped remotely. The App may not receive acknowledgment that the pump stopped, which is important device state information to confirm safe operation. It may be that the pump was not designed (legacy pump) or configured to acknowledge the stop command or that the pump has malfunctioned. Examples of other important state information could include: the operator may halt the pump completely instead of returning to a ready state; the pump motor may require a full revolution to stop instead of stopping immediately, thereby delaying an immediate stop request. These are examples of the interaction being dependent on the state of the component, the pump, receiving the signal [15].

Other factors to consider in a device model, related to state and context of use may be: How long does pump take to shut off? Was the pump in the correct mode of operation such that it could be shut off from another system component? Is the pump required to send a message back to the controller that it is shutoff to confirm its actions? All possible pump scenarios need to be considered in order to construct a well formed contract and device model.

Example 3: Finger pulse oximeter and Blood Pressure (BP) cuff (same arm). The device model does not know the context (e.g. BP cuff proximal to SpO_2 probe).

Consider the situation of a patient with an automated non-invasive blood pressure (NIBP) cuff placed on the arm and a pulse oximeter finger probe distal to that cuff on the ipsilateral arm. The periodic inflation of the cuff impedes both arterial and venous flow from that arm and as such causes intermittent saturation and pulse rate errors associated with cuff inflation. The impact of this cuff-induced hypoperfusion [16] on the time of the disappearance of displayed saturation values is dependent on the particular pulse oximeter (i.e. its device model). Communicating the state of the non-invasive blood pressure monitor to the pulse oximeter in order to indicate that the blood flow is being perturbed could help prevent erroneous data from being displayed and reported, thereby potentially preventing erroneous clinical diagnoses (IB 3/4).

Example 4: *EHR and medical device data. The device model is implicit on receiving end.*

Consider the transfer of physiologic data from a bedside monitor to the electronic health record (EHR). What actually is recorded and how it is recorded (frequency/format) is dependent on the settings in the physiologic monitors. Relevant portions of the device model in the form of metadata, such as configuration settings for signal filters and alarm values, are rarely transmitted (IB 1). Nor is a time stamp indicating when the data was acquired. Typically, the time stamp within the record reflects time of receipt. As a result of this missing information, when the waveforms are recalled for later viewing, the absolute time and time alignment between the signals is unknown. As in Example 1, not knowing the settings of the waveform filter that was applied during the data collection may prevent or hinder definitive conclusions being drawn about certain clinical features (e.g. rapid desaturations, ST segment analysis) [17]. The data retrieval and interpretation problem is further complicated by having waveforms saved in separate data stores, and possibly not readily available, as the information in the waveform not extracted earlier as parameters may be lost. Even if the data was stored in the same data store but was collected as part of a multi-center study, the implicit device model from each center using different brands or models of the same type of monitor may not be consistent. In this case, the data cannot be reliably pooled, analyzed or compared without shared semantics.

Example 5: X-Ray/Cardiopulmonary Bypass, and ventilation or Oxygen and Laser use. The device models do not share operational states.

Consider the situation during surgery when ventilation therapy must be temporarily interrupted for a short procedure; for example, the need to synchronize the interruption of supplemental oxygen delivery with the operation of a CO_2 laser, so as to minimize the chance for an airway fire. Even with ''laser safe'' tubes, it is recommended to maintain the minimum inspired oxygen concentration necessary (e.g. close to room air) during the operation of laser within the airway [18]. With an interoperable system, the laser system should not turn on until it gets a message from the oxygen delivery device that its state is OFF (IB 3/4) or that inspired oxygen is at a safe level. Similarly, the oxygen concentration should not increase until its gets a message confirming that the laser state is off.

A similar level of coordination or synchronization supporting safety is required for an abdominal or chest X-ray performed during surgery on a ventilated patient. The procedure may require the temporary cessation of ventilation to prevent motion-induced image artifact or to time the exposure with the breath cycle. In the case of a cholecystectomy (gall bladder removal) with intraoperative cholangiography (x-ray), the ventilation is stopped and the cholangiogram is performed with contrast to identify internal structures. Depending on the procedure, the synchronization with the imaging can be with either expiration or inspiration. A group at the University of Florida prototyped a system to synchronize the image with the end of inspiration to ensure that the images are obtained at maximal lung inflation; thereby improving the quality of the radiograph. [19]. In order to obtain a clinically relevant image, the X-ray should not record an image until it receives a signal that ventilation is suspended. Similarly, ventilation should not resume until it receives a signal that the X-ray procedure has been completed or ventilation should resume automatically on its own if the signal is not received within a defined time interval (IB 3/4). Manually disabling ventilation for an X-ray procedure followed by unanticipated equipment problems have led to excessive delays in the resumption of patient ventilation with unfortunate outcomes [20]. An interoperable platform-based approach to this failure to ventilate has been prototyped [21]. Although not available commercially, enabling device capabilities have been incorporated in the latest Anesthesia Workstation and Critical Care Ventilator standards. This coordination requires a device model include the clinical state of the patient (mechanical pulmonary ventilation) and the state of the device (inhalation, exhalation, pause). Analogous to the X-ray-ventilation scenario is the situation with the temporary cessation of ventilation associated with the use of cardiopulmonary bypass. An excessive delay in the resumption of ventilatory support post-bypass can be life threatening or result in major organ damage [22]. In the scenarios highlighted, integration of the devices into an integrated, networked system could improve patient safety by the sharing of operational states between devices thereby minimizing the likelihood of a failure to ventilate or deliver higher levels of oxygen during airway laser procedures.

X. DISCUSSION

Components within a system need to have an equivalent understanding of the interactions that occur and the information exchanged in order to function safely. Differences between the model for the sender and receiver can lead to hazardous situations (e.g. R2 not compatible with S1). This may or may not have clinical ramifications depending on the intended use of the system. For example, manufacturer A may choose to display vital sign trends on their front panel display using data collected from the output of a different multi-parameter monitor. Manufacturer A establishes the characteristics of the signals needed, as stated in the labeling. For some of the display ''trend'' parameters, there may be accepted standards or guidelines such that the signals can be used and interpreted properly. Wide spread adoption of legacy industry standards (e.g. safety/performance or informatics standards) may allow users to reach a consensus about the implicit device model. The opposite situation may exist with closed loop control, as in the case of manufacturer B controlling ventilator settings (e.g. optimizing FiO_2 and/or PEEP) based on parameters derived from the pulse plethysmogram (e.g. SpO_2 values) as measured with a pulse oximeter. In this case, each manufacturer has their own ''private'' device model hence there is no widely accepted standard for the pulse plethysmogram data, nor is there a standard model for the response time of an oximeter.

In the case of the above manufacturer A, the receiver or consumer of the data, (e.g. the trend or EHR vendor) may spend considerable effort to assure reliable communications from the physiologic monitors while assuming that the underlying signals (e.g. ECG, NIBP, SpO_2) are sufficiently standardized and accurate and as such that revalidation of the clinical performance is not needed. This is an example of where the contract is implicit [21].

If the sender does not intend to change the state of the receiver (e.g. populating the EHR), a manufacturer's risk analysis has less complexity than if the intent is to change the state, as the consequences of the state changes have to be considered for the latter. If the receiver intends to change its own state based on the received data, a more complex risk analysis is required as it needs to include the possible failure modes of the sending component.

Many medical devices fall within the scope of point-of-care (POC) technologies. These include POC laboratory tests, vital sign measuring technologies, and infusion pumps. This paper has described a conceptual framework with relevant clinical examples to emphasize the importance of and role of state and context in the design of medical devices. This is particularly crucial for devices used at the POC where context determines the requirements for the device model in terms of available parameters and device capabilities. Even though unique device identifiers and control of access to the device information would be included in a component's device model, matters such as operator identification and authentication are related more to system-level design aspects and can serve as a means for verifying that the correct components are being used, rather than determining correct device behavior. An interesting challenge with POC technology can arise when non-experts are involved in applying medical sensors. For example, misapplication of a blood pressure cuff could cause over or under-reading of blood pressure values. Therefore, the validity of data may be influenced by the state (correct or incorrect blood pressure cuff size for that patient) and clinical context (patient active during measurement).

XI. CONCLUSION

In the paper, the authors have identified the different kinds of information that need to be included in a contract between system components in the form of a device model (e.g. state, context, change of either) and have discussed where this information comes from, including approaches to capture information and why the information in the device model is necessary for safety. The paper has described an approach to achieve safe medical device PnP using comprehensive device models that are ''standardized''-characterized by dynamic interoperability between devices and conceptual interoperability within the organization utilizing the levels of interoperability concepts of Tolk and Muguira [1], Turnitsa [2], and Robkin *et al.* [3]. And lastly the paper proposes that shared clinical scenarios enable developers to be conceptually interoperable, sharing the same goals, constraints, and processes which is necessary for their work product to achieve dynamic interoperability. Future research efforts will be aimed at: developing a universal device model to enable dynamic assembly of clinical applications from well-known and characterized functional components; to enable device models to be re-used; and to architect safe communication between system components. Ranganath *et al.* [23] identify a collection of common communication patterns used in distributed systems including publisher-subscriber, requester-responder, sender-receiver, initiator-executor, and orchestration patterns. These patterns can be viewed as more detailed versions of the Interoperability Behaviors presented in Section IV. In addition, Ranganath *et al.* [23] propose quality of service contracts for each pattern that provide a basis for applications and devices to specify real-time communication constraints that be automatically checked at integration time and runtime. Publicly available research implementation frameworks illustrate how the patterns can be implemented using several widely-used middleware frameworks. The relevance and urgency of such an approach is increasing, particularly with point-of-care monitoring and therapeutic devices, given the ongoing changes to healthcare delivery models and precision medicine initiatives.

Application developers will have a hardware agnostic platform to host their devices. Sensor and actuator manufacturers will be able to provide greater capabilities and flexibility to application developers to use their services in novel ways to improve the health and safety of patients. In order to properly implement such a system requires the necessary upfront systems engineering efforts so that proper system requirements specification, design and validation can be performed. Kim *et al.* [24] propose high-level requirements for the ICE Device Model that addresses requirements on information content, communication patterns, quality of service, security, safety, behavioral specifications, as well as requirements on device model authoring and compliance evaluation tools. A robust device model is key to enabling seamless device interoperability.

ACKNOWLEDGMENTS

We like to acknowledge the contributions to this work of Steven Dain, MD from the University of Waterloo, Canada, Jennifer Jackson, formerly with the Brigham and Women's Hospital and MD PnP Program, now with the Cedars-Sinai Medical Center, Los Angeles, CA and David Osborn, Sr. Manager of Global Regulations & Standards Philips.

REFERENCES

[1] A. Tolk and J. A. Muguira, ''The levels of conceptual interoperability model,'' in *Proc. Fall Simulation Interoper. Workshop*, Orlando, FL, USA, 2003.

[2] C. D. Turnitsa, ''Extending the levels of conceptual interoperability model,'' in *Proc. IEEE Summer Comput. Simulation Conf.*, 2005.

[3] M. Robkin, S. Weininger, B. Preciado, and J. Goldman, ''Levels of conceptual interoperability model for healthcare framework for safe medical device interoperability,'' in *Proc. IEEE Symp. Product Compliance Eng. (ISPCE)*, May 2015.

[4] J. M. Goldman, ''Solving the interoperability challenge: Safe and reliable information exchange requires more from product designers,'' *IEEE Pulse*, vol. 5, no. 6, pp. 9–37, Nov./Dec. 2014. [Online]. Available: http://pulse.embs.org/november-2014/solving-interoperability-challenge/

[5] HITI, ''Medical device interoperability,'' AAMI, Arlington, VA, USA, White Paper AAMI MDI/2012-03-30, 2012.

[6] *AAMI Association for the Advancement of Medical Instrumentation Medical Electrical Equipment—Part 2-4: Particular Requirements for the Safety of Cardiac Defibrillators (Including Automated External Defibrillators)*, document ANSI/AAMI DF80:2003, 2003.

[7] *Agilent Series 50 Fetal Monitors, Digital Interface Protocol Specifications*, document M1350-9014B, Agilent Corporation, Waldbronn, Germany, Jul. 2000.

[8] *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*, document ASTM F2761-09, 2013.

[9] *Information Technology—Open Distributed Processing—Reference Model: Foundations*, document ISO/IEC 10746-2, 2009.

[10] *Industrial Communication Networks—Fieldbus Specifications—Part 1: Overview and Guidance for the IEC 61158 and IEC 61784 Series*, document IEC 61158-1, 2014.

[11] *Version 3 Standard: Reference Information Model, Release 1*, document ANSI/HL7 V3 RIM, R1-2003, Ann Arbor, MI, USA, Health Level Seven International, 2003.

[12] S. Dain, T. Rausch, and J. M. Goldman, ''Domain information model for the patient centric integrated clinical environment (ICE DIM),'' in *Proc. Conf. Soc. Technol. Anesthesia Annu. Meeting*, 2015, pp. 1–2.

[13] *Health Informatics—Point-of-Care Medical Device Communication—Part 10201: Domain Information Model*, ISO/IEEE document 11073-10201, 2004.

[14] S. Zafar, I. Ayappa, R. G. Norman, A. C. Krieger, J. A. Walsleben, and D. M. Rapoport, ''Choice of oximeter affects apnea-hypopnea index,'' *Chest*, vol. 127, no. 1, pp. 80–88, Jan. 2005.

[15] D. Arney, R. Jetley, P. Jones, I. Lee, and O. Sokolsky, ''Formal methods based development of a PCA infusion pump reference model: Generic infusion pump (GIP) project,'' in *Proc. Joint Workshop High Confidence Med. Devices, Softw., Syst. Med. Device Plug-Play Interoper.*, Jun. 2007, pp. 23–33.

[16] T. Kawagishi, N. Kanaya, M. Nakayama, S. Kurosawa, and A. Namiki, ''A comparison of the failure times of pulse oximeters during blood pressure cuff-induced hypoperfusion in volunteers,'' *Anesthesia Analgesia*, vol. 99, no. 3, pp. 793–796, Sep. 2004.

[17] J. Zaleski, *Integrating Device Data into the Electronic Medical Record.* Erlangen, Germany: Publicis, 2009.

[18] S. Roy and L. P. Smith, ''Surgical fires in laser laryngeal surgery: Are we safe enough?'' *Otolaryngol. Head Neck Surg.*, vol. 152, pp. 67–72, Jan. 2015.

[19] P. B. Langevin, V. Hellein, S. M. Harms, W. K. Tharp, C. Cheung-Seekit, and S. Lampotang, ''Synchronization of radiograph film exposure with the inspiratory pause. Effect on the appearance of bedside chest radiographs in mechanically ventilated patients,'' *Amer. J. Respirat. Crit. Care Med.*, vol. 160, pp. 2067–2071, Dec. 1999.

[20] A.S Lofsky, “Turn your alarms on, APSF Newsletter,” *Winter*, vol. 19, no. 4, p. 43, 2005.

[21] D. Arney *et al.*, “Design of an X-ray/ventilator synchronization system in an integrated clinical environment,” in *Proc. Conf. IEEE Eng. Med. Biol. Soc.*, Aug./Sep. 2011, pp. 8203–8206.

[22] R. A. Caplan, M. F. Vistica, K. L. Posner, and F. W. Cheney, “Adverse anesthetic outcomes arising from gas delivery equipment: A closed claims analysis,” *Anesthesiology*, vol. 87, pp. 741–748, Oct. 1997.

[23] V.-P. Ranganath, Y. J. Kim, J. Hatcliff, and Robby, “Communication patterns for interconnecting and composing medical systems,” in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, pp. 1711–1716.

[24] Y. J. Kim, J. Hatcliff, V.-P. Ranganath, Robby, and S. Weininger, “Integrated clinical environment device model: Stakeholders and high level requirements,” in *Proc. Med. Cyber Phys. Syst. Workshop SIGBEG Rev.*, 2015, pp. 1–10.

SANDY WEININGER received the B.S.E.E. and M.S./B.M.E. degrees from Drexel University with a focus on the properties of the electrode-tissue interface, and the Ph.D. degree in bioengineering from the University of Pennsylvania with a specialization in signal processing and control systems related to behavioral assessment of newborns. His current scientific and regulatory focus is the performance assessment of sensors and actuators for physiologic systems, and identifying interactions within complex systems to assess safety. He works on standards development organizations, including UL, AAMI, ASTM, IEC, and ISO to construct both horizontal and vertical safety standards. He is actively involved in developing and delivering courses on achieving safety in medical devices using systems engineering principles. He is a member of AAMI/UL 2800—Interoperable Systems, AAMI Interoperability Working Group, and was the Chair of the ASTM Pulse Oximeter Committee and FDA’s liaison to IEC TC 62 and SC 62A, committees responsible for safety of electromedical equipment, and ISO TC 121/SC3 JWG10 - oximeters. He works with academic partners to develop methods for the evaluation of interoperable systems.

MICHAEL B. JAFFE (SM’77–M’82–S’12) received the B.S. degree from Cooper Union for the Advancement of Science, the M.S. degree from Dartmouth College, and the Ph.D. degree in biomedical engineering from the University of Southern California in 1994. He currently consults on medical device technologies and health IT. He has worked since 1981 in the respiratory field, working for such companies, such as Beckman Instruments, Sensormedics, Respironics, and Philips. He has served as a Co-Editor of *Capnography: Clinical Aspects* (Cambridge University Press, 2004). He has served as the Secretary of the IEC/ISO Joint Working Group for ISO 80601-2-55, respiratory gas monitors. He is also a member of several international standards committees relating to anesthesia and respiratory equipment and a member of the Anesthesia Patient Safety Foundation Committee on Technology. He has authored 18 peer-reviewed publications and holds over 30 U.S. patent families in patient monitoring.

MICHAEL ROBKIN is the President of Anakena Solutions, Inc. He has been developing cutting-edge, mission-critical computer applications for more than 20 years as a Programmer, a Systems Architect, and an Executive at Hughes Aircraft, GM-Europe, and Kaiser Permanente. He was a Founding Member of the Board of Directors and a Treasurer of the Continua Health Alliance.

TRACY RAUSCH received the B.S. degree in biomedical engineering from Wright State University and the M.S. degree in mechanical engineer from The Catholic University of American. She is the CEO and Founder of DocBox, Inc. She is a Certified Clinical Engineer. She has been serving as an Advisor to the MDPnP program since 2006.

She has interest in the integration of technology into clinical environment and the safety and process improvements in patient care.

DAVID ARNEY is the Lead Engineer for the Medical Device Plug and Play Program. He has been working on applying formal methods to medical device software since 2001 and was a Scholar in residence with the FDA’s Center for Devices and Radiological Health in the Office of Science and Engineering Laboratories. He was involved in writing the ASTM 2761-09 ICE standard for interoperable medical devices. He started at the MD PnP program in 2010, and is currently writing his dissertation for a Ph.D. in computer science under Professor Insup Lee with the University of Pennsylvania.

JULIAN M. GOLDMAN is a Medical Director of Biomedical Engineering for the Partners HealthCare System, the Founder and Director of the Medical Device “Plug-and-Play” (MD PnP) Interoperability program, and an Anesthesiologist with Massachusetts General Hospital (MGH). He performed his clinical and research training at the University of Colorado, and is Board Certified in Anesthesiology and Clinical Informatics. He served as a Visiting Scholar in the FDA Medical Device Fellowship Program as well as a Chief Medical Officer of a medical device company. At MGH, he served as a Principal Anesthesiologist in the “OR of the Future.” He founded the MD PnP program in 2004 to promote innovation in patient safety and clinical care by leading the adoption of patient-centric medical device integration. He currently serves in leadership positions in several medical device standardization organizations, including the Chair of the ISO Technical Committee 121, the Co-Chair of the AAMI Interoperability Working Group, and the Co-Chair of the Healthcare Task Group of the Industrial Internet Consortium. He is the former Chair of ASTM F29 and of the ASTM subcommittee that developed the ICE standard. He is an IEEE EMBS Distinguished Lecturer.