Challenges of distributed risk management for medical application platforms

Conference Paper · May 2018

DOI: 10.1109/ISPCE.2018.8379270

CITATIONS
9

READS
470

4 authors:

John Hatcliff
Kansas State University
197 PUBLICATIONS **5,419** CITATIONS
SEE PROFILE

Eugene Y. Vasserman
Kansas State University
52 PUBLICATIONS **936** CITATIONS
SEE PROFILE

Todd Carpenter
Adventium Labs
12 PUBLICATIONS **61** CITATIONS
SEE PROFILE

Rand Whillock
Adventium Labs
22 PUBLICATIONS **28** CITATIONS
SEE PROFILE

Challenges of Distributed Risk Management for Medical Application Platforms

John Hatcliff, Eugene Y. Vasserman
Kansas State University

Todd Carpenter, Rand Whillock
Adventium Labs

***Abstract*— ISO 14971, the primary medical device risk management standard focuses on single-manufacturer monolithic devices. However, the trend towards medical systems built from reusable platforms and interoperable components produced by different manufacturers introduces a number of additional risk management challenges.**

In this paper, we revisit the stages of the ISO 14971 risk management process, identify risk management challenges associated with interoperable medical systems that are not sufficiently addressed in ISO 14971, and we discuss possible process, analysis, and management concepts that may be useful in addressing these challenges.

I. Introduction

Risk Management is an overarching collection of activities central to development, assurance, and regulatory submissions of medical devices. Historically, risk management (e.g., as presented in the primary medical device risk management standard ISO 14971) has been concerned with identifying potential *safety-related* problems associated with a device then designing and verifying "controls" to prevent harm to patients and operators.

Over the last 5-10 years, the scope of medical device risk management has been expanding to include additional concerns related to placing devices in the context of hospital networks (as reflected e.g., in IEC 80001) and security risks (as reflected e.g., in AAMI TIR 57 and UL 2900). IEC 80001 provides some initial directions for the sharing of risk-related information in the form of "disclosures" between device manufacturers and health delivery organizations (HDOs). In the IEC 80001 context, HDOs have the responsibility of performing risk management on the integration of medical devices into the broader hospital IT network. IEC 80001 also incorporates security as a "top-level property" and a driver of risk management activities. AAMI TIR 57 addresses pre-market issues in single medical devices (as opposed to integrated systems) and provides suggestions for how security risk management can be interleaved with safety risk management [2]. UL 2900 (and its specialization for medical devices in UL 2900-2) focuses on testable requirements for appropriate security engineering and technologies for single network-enabled medical devices. In summary, existing standards and guidance tend to focus on *safety* risk management of *single* devices (ISO 14971), security of *single* devices or incorporation of *single* medical devices into a hospital IT network (IEC 80001) where the primary risk management of integration lies with HDO.

This work was supported in part by the US National Science Foundation (NSF) FDA Scholar-in-Residence award CNS 1565544, by US Army Medical Research and Materiel Command under Contract #W81XWH-16-C-0192, and the Department of Homeland Security (DHS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency (HSARPA), Cyber Security Division (DHS S&T/HSARPA/CDS) BAA HSHQDC- 14-R-B0005, the Government of Israel and the National Cyber Bureau in the Government of Israel via contract number D16PC00057.

A. Trends and Challenges

Medical devices are increasingly being built using interoperability and platform approaches that enable devices and service components to be combined flexibly into "systems of systems" to achieve integrated care-giving solutions that typically exceed the capabilities of stand-alone devices. Work in the research [14], [28], [13], [25], [42] and standards [5] communities is laying the foundations for safety, security, and risk management approaches for "systems of systems" of medical devices built using "medical application platforms" (MAP). A MAP is a safety- and security- critical real-time computing platform for (a) integrating heterogeneous devices, medical IT systems, and information displays via a communication infrastructure and (b) hosting application programs ("apps") that provide medical utility via the ability to both acquire information from and update/control integrated devices, IT systems, and displays. Consortia [29], [18] are being organized to help support ecosystems of manufacturers that cooperate to build asset bases of reusable components and rapid system development approaches aligned with a particular architecture.

In this context, the challenges of risk management are expanding and stressing current medical risk management frameworks presented in ISO 14971 and IEC 80001. We sketch some of the challenges below that we address in greater detail in the sections that follow.

Instead of being focused in a *single* standalone device manufacturing organization, risk management activities, including component-level and system-level hazard analysis tasks, must cross boundaries between organizations that build components and/or integrate components into systems. At the system-level, this will require integrating the different risk management strategies, assumptions, and results across different categories of system elements including conventional medical devices, infrastructure components (which may originate in the IT community and have not been developed following medical quality management or risk management processes), software-based medical application logic, and reusable safety, security, and medical service components such as forensic data loggers [4]. Thus, risk management

standards and industry guidance documents can not confine their primary attention to conventional hardware-oriented medical devices. Because components are now often used in multiple system contexts, component manufacturers will need to carry out hazard analysis and implement risk controls without knowing the specific system context into which their products will be integrated. This makes it difficult to identify specific patient harms and hazardous situations as required by ISO 14971. A key element of the interoperability vision is that reuse is not just limited to component implementations – the component's risk management and assurance results should also be reusable when the component is integrated into a new system context. Without this broad concept of reusability, component-level assurance and risk management would need to be redone every time a component is integrated into a different system context. It follows that the results of risk analysis and risk control activities will need to be communicated (via different different forms of "hand-offs" or transactions) across organizational boundaries to system integrators in such a way that the results are trustworthy, but limit the disclosure of proprietary information to any extent possible. System integrators will need to acquire results from component manufacturing organizations that may have disparate processes, analysis, reporting styles, and reuse the component-level risk management results (without knowing exact details of component implementations) to support system-level risk management and safety arguments.

Broader attack surfaces introduced by interoperability mechanisms will need to be accounted for in risk analysis that addresses how security vulnerabilities may impact safety [34]. When security controls are introduced, analysis will need to consider how improved security might negatively impact the ability of context entities (operators or other interoperable components) to access in a timely fashion capabilities needed for patient safety.

To address issues of responsibility and liability that arise in this new decentralized development paradigm, processes, meta-data schemes, and execution-time monitoring and logging need to be developed to allocate responsibility of analysis and risk controls across components from different manufacturers and to establish provenance and accountability in the presence of adverse events. Current management processes (e.g., in ISO 13485) do not adequately specify cross-organization communication processes to communicate post-production quality issues and adverse event reports. Such processes would enable system/component manufacturers to understand how the impacts of problems in depended-on components need to be addressed.

B. Contributions of this paper

In this paper, we expand on the challenges outlined above and present solution directions for moving towards a distributed risk management approach needed for interoperable medical systems. Our proposed solution directions will often include suggestions for how emerging medical device interoperability standards might augment certain clauses of ISO 14971 with new requirements that address challenges specific to interoperability and multi-organization development. In this work, we draw from our observations and experience in implementing prototypes of medical application platforms [26], [9], [37], working with manufacturers of interoperable platforms and devices [10], [6], [29], [22], working on safety and security standards for interoperable medical systems [41], [19], [2], building tooling to support risk analysis for interoperable medical systems [30], and working on safety analysis in other domains including avionics.

While medical device risks associated with electrical, chemical, and biological hazards are important, the problems are mature and well-studied, and largely orthogonal to issues associated with interoperability. In this paper we limit the discussion to issues surrounding functional safety, security, and essential performance.

The concepts in this paper are applicable to single devices with interoperability capabilities, but we are most interested in advancing solutions in the context of multi-device / multi-vendor engineered medical device interoperability frameworks (including platform approaches) that may be supported by consortia. In this context, we see an opportunity to "pre-coordinate" many aspects of risk management across platform design, development, and consortia member activities so as to avoid offloading *all* integration risk management to HDOs to be addressed as part of IEC 80001-covered activities. We argue that it is medical device and platform manufacturers that have the deepest product technical knowledge and strongest systems engineering competencies for interoperability and integration, and that the medical device community needs to establish standards that facilitate manufacturer-centric intra-platform integration risk management (instead of leaving this responsibilities to HDOs) while focusing HDO responsibilities on health IT integration at platform boundaries and alignment of pre-coordinated platforms risk management with HDO organizational and care-giving policies. Finally, we envision third party certification to emerging standards such as AAMI/UL 2800 as playing a key part of ensuring trust in risk management results between manufacturers.

Note that although the challenges in this paper are presented in the context of the medical domain, they are also applicable to a wide range of cyber-physical system domains including avionics, automotive, the industrial internet, and the Internet of Things (IoT) in general.

Section II summarizes properties of an example reference architecture for MAPs and associated development ecosystem. Section III discusses a simple example interoperable medical system that instantiates the architecture. Section IV provides background on the medical device risk management processes and terminology. Section V presents challenges in supporting risk management for interoperable systems, organized according the primary sections of ISO 14971. Finally, Section VI states conclusions and gives some directions for future work.

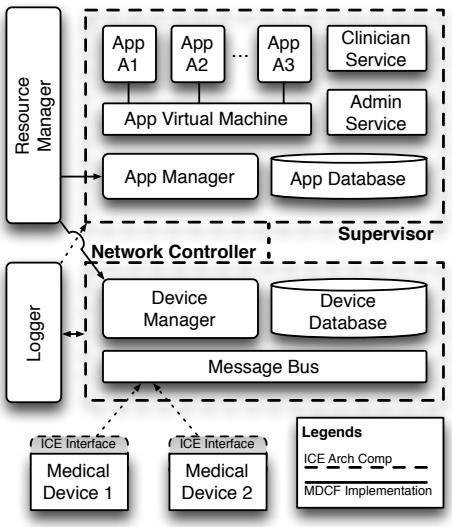

Fig. 1. ICE Architecture and MDCF Realization

II. Medical Application Platform Example

MAPs can be realized via different architectures. The Integrated Clinical Environment (ICE) [5], standardized in the ASTM F2761-2009, is one such architecture; ICE development has been led by the CIMIT Medical Device Plug-and-Play (MD PnP) interoperability project. ASTM F2761-2009 identifies the primary architectural components of ICE and their functionality as it relates to the MAP goals of interoperability and safety. The US Food and Drug Administration (FDA) recognizes it as a medical device interoperability standard [12]. The emerging ICE Alliance consortium [18] is facilitating vendor cooperation and the development of implementation ICE standards. Related interoperability approaches include the OR.NET consortium [29] that has spear-headed the newly developed 11073 Service-oriented Device Communication (SDC) standards [19] and has produced interoperability infrastructure implementations that have been used in multiple clinical technology demonstrations [21], [20].

In the following sections, we will often use generic architecture terminology when the discussion does not depend on a particular architecture. We use *item* to describe both an interoperable system as well as interoperable components such as medical devices, apps, or infrastructure components. Thus, item implementations may be constructed by integrating sub-items or they may not decompose further with respect to interoperability (they represent the lowest level of interoperability hierarchy). Note that these generic definitions are recursive (an item may be implemented using other sub-items, etc.) to support discussions of system elements with arbitrary interoperability hierarchy.

A. An Example MAP Architecture and Realization

The boxes with dashed lines in Figure 1 present the ICE architecture. ASTM F2761 [5] only identifies primary components such the Supervisor, Network Controller, etc. using short 3–5 sentence descriptions, and gives no detailed requirements nor implementation strategy for any of these components. Thus, there have been significant efforts within the research community to investigate appropriate requirements [16], [14] implementation technologies [26]. In joint work between Kansas State University, University of Pennsylvania, and FDA engineers, a prototype ICE implementation called the Medical Device Coordination Framework [26] was developed, and we use concepts from the implementation to make the discussion of the ICE architecture more concrete. The boxes with solid lines in Figure 1 indicate how MDCF components realize the ICE architecture.

Network Controller: The Network Controller provides a high-assurance network communication capability, establishing virtual "information pipes" between devices and apps running in the Supervisor. It exposes the ICE interfaces of attached devices to Supervisor apps, and is agnostic to the intended use of the clinical apps.

The MDCF implements the ICE Network Controller as a collection of services. The *Message Bus* abstracts out the low-level networking implementation (e.g., TCP/IP) and provides a high-level publish/subscribe messaging service. All communication between medical devices and the MDCF occurs via the Message Bus, including protocol control messages, patient physiologic data, and commands. It also provides basic real-time guarantees (e.g., bounded end-to-end message transmission delays) that apps can state as requirements and then take as assumptions. Additionally, the Message Bus supports various fine-grained per-pipe security policies. The *Device Manager* implements the server side of the MDCF device connection protocol (medical devices implement the client side) and tracks the connectivity of those devices, notifying appropriate apps if a device goes offline unexpectedly. The Device Manager also validates the trustworthiness of connecting devices by verifying their machine-readable credentials to determine if the device is genuine (i.e. not counterfeit) and has been cleared to operate in the current network environment, e.g., does not produce more data than the network can support.

Supervisor: The ICE standard states that the Supervisor "provides a platform for functional integration between ICE compliant equipment via the network controller and can provide application logic and an operator interface" [6], but does not explicitly identify the notion of an "app"; the mechanism through which the "application logic" can be programmed, organized, or supported by an execution environment that provides appropriate guarantees is missing completely.

One of the contributions of our work has been to flesh out a vision and initial implementation of the supervisor/app concept. In our view, the Supervisor should be thought of as a virtual machine that hosts *Supervisor Apps*. It should provide separation/isolation kernel-like [35] data partitioning (preventing information leaks between apps, and apps cannot inadvertently interfere with one another) and time partitioning (real-time scheduling guarantees that one app's computation or memory requirements cannot catastrophically affect the performance of another app). In support of this, the

current MDCF implementation provides a virtual machine in which apps execute. Each app declares *device types* indicating the types/capabilities of devices upon which it depends. When a clinician launches the app, the Supervisor queries the Network Controller to determine if a device that meets those requirements is currently on the network and associated with the patient under consideration. If more than one device satisfies the requirement, the operator chooses a particular device to bind to the app.

The *Data Logger* records for forensic purposes important system events and information exchanged between components. This recorded data can then be played back after system failures to assess the cause (this is similar to the role of, e.g., the "black box" in aircraft).

B. MAP Implementations

DocBox [10] provides a commercial ICE platform, while the Medical Device Plug and Play group platform provides an open-source prototype based on DDS called OpenICE [28]. with the ICE architecture, The Medical Device Coordination Framework (MDCF) (e.g., [26]) developed by researchers at Kansas State University and the University of Pennsylvania was one of the earliest open-source ICE prototypes. Components added to the MDCF are presented in solid-lined boxes in Figure 1. The MDCF provides a middleware substrate and associated services [25], tools for authoring apps, generating executable APIs [32], [23], and performing risk management activities [31].

Several different implementations [40], [39] have been provided for the OR.NET [22] platform. While the OR.NET architecture can be aligned with ICE as presented in ASTM F2761, it places greater emphasis on distributed interactions (as opposed to actions centrally coordinated by an ICE Supervisor) inspired by Service-Oriented Architecture (SOA) approaches. Component interfacing OR.NET is achieved using the recently developed 11073 SDC standards (see [8] for illustration) – an enhancement of the legacy IEEE 11073 medical device interoperability standards.

C. MAP Ecosystems

The risk management challenges addressed in the paper are applicable to any of the approaches or implementations above as well as approaches championed by other organizations including the Center for Medical Device Interoperability [1]. For any applicable context, we assume the ability to characterize the *interoperability ecosystem* in which the platform asset base is developed and applied. Kim et al. [23] define an interoperability ecosystem as as the collection of stakeholders, artifacts that are produced, processes that are followed, and trust relationships that are established, to develop, assure, market, deploy, and operate interoperable systems. For MAPs, the interoperability ecosystem should lead to trustworthy components and to the safety and security of medical systems built using ecosystem assets.

A set of development organizational roles with clearly identified activities and responsibilities aligned with the ecosystem's reference architecture is a important element in an ecosystem management. In the ICE ecosystem, one might recognize device development, app development, platform development, and system development as roles with lifecycle activities that involve some degree of coordination with other roles. For example, a device developer may coordinate with a platform developer to build an interoperable device with middleware protocols and data models that align with those of the platform and that conforms to a disclosed interface specification that defines the capabilities of the device available through its interoperability interface. A platform developer may produce an implementations of the network controller and supervisor and provide resources that facilitate the development and assurance of devices and apps that use the platform. An app developer may build a software app that provides care-giving functionality via application logic running on the platform and interactions with interoperable devices attached to the platform. In the ICE context, the app developer will typically take on system engineering responsibilities as well; since an app defines the medical intended use of a particular configuration and instantiation of the platform, the app manufacturer must be able to assure end-to-end safety/security properties about behavior that results when executing the app on a platform with a specific set of devices [17].

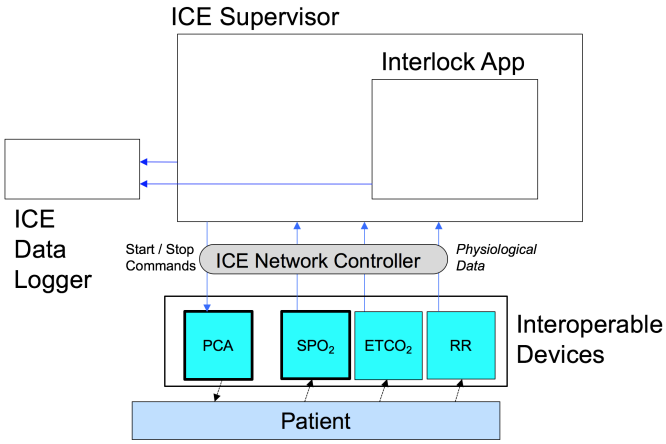

Fig. 2. PCA Monitoring and Safety Interlock System

III. PCA Safety Interlock Example

A Patient-Controlled Analgesia (PCA) pump is a medical device often used in clinical settings to intravenously infuse pain killers (e.g., opioids) at a programmed rate into a patient's blood stream. A PCA pump also includes a button that can be pushed by the patient to receive additional bolus doses of drug – thus allowing patients to manage their own pain relief. Despite settings on the pump that limit the total amount of drug infused per hour and that impose lock out intervals between each bolus dose, there is still a risk of overdose when using PCA pumps.

Figure 2 illustrates how an ICE-based MAP can be used to implement an interoperable medical system (referred to as the PCA Monitoring and Safety Interlock System (PCA MSIS)) in which an PCA Infusion Monitoring and Safety

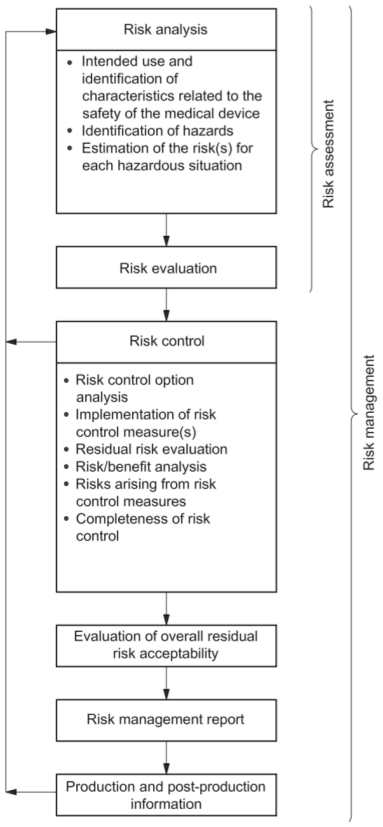

Fig. 3. ISO 14971 Risk Management Overview.

Interlock App (short: Interlock app) that integrates with monitoring devices to obtain physiological parameters such as blood oxygen saturation (SpO2), End-Tidal carbon dioxide (EtCO2), and respiratory rate (RR) from monitoring devices that are useful for monitoring for respiratory depression (an indication of PCA overdose). The app implements a safety interlock by halting the pump's opioid infusion (via its network interface) when the monitored physiological parameters satisfy a *halt condition*, i.e., have values and/or trends that may signal the onset of a respiratory depression. When a *halt condition* is detected, the Interlock app also provides both a visual and audio alarm through the user interface of *Supervisor*. The PCA MSIS is designed for interoperability, e.g., monitoring devices from different manufacturers can be integrated, as long as their network-exposed capabilities are compliant with an ICE Device Specification that satisfies the app's requirements.

We have chosen this example to illustrate risk management concepts because it has been subject of a number of demonstration in the ICE community (see e.g., [3], [24]) as well as a subject of current standardization activities.

IV. Risk Management Background

Figure 3 illustrates the primary steps medical risk management as specified by ISO 14971 risk management process. We summarize these activities below while slightly modifying the arrangement of steps so as to enable a conceptually cleaner summary of interoperability challenges and directions.

- Scoping: Statement of the device's purpose (intended use) and description of characteristics of the device related to safety (part of 14971 Risk Analysis),
- Risk Analysis: Identification of hazards, analysis to determine hazardous situations capturing causal factors that may lead to hazards, and estimation of risk for each hazardous situation,
- Risk Evaluation: Evaluation of risk analysis results to determine the hazardous situations for which risk controls will be realized,
- Risk Controls: Design and implementation of measures to mitigate hazardous situations including eliminating causal factors or hazards through design, achieve robustness to causal factors through fault tolerance techniques, detecting hazard situations and moving the device to a safe state or notifying operators that action is needed to move the device to a safe state, and information (e.g., operator instructions) on steps that the operating organization must take to support safety,
- Risk Control Verification and Evaluation: Verification that risk controls for each hazardous situation are implemented correctly and are effective, evaluation to determine if risk controls achieve an acceptable level of residual risk, and analysis to determine if the controls themselves introduce new risks,
- Evaluation of Overall Residual Risk Acceptability: Overall evaluation across all hazardous situations to determine if residual risk is acceptable,
- Risk Management Report: Production of a risk management report documenting risk controls as well as analysis and evaluation results,
- Production and Post-production: Monitoring and tracking safety-related problems of devices as they are operated in the field and continued evaluation of the effectiveness of the risk controls in light of the discovered problems.

V. Risk Management Challenges

In this section, we present a summary of challenges for risk management in the context of medical application platforms following the outline of ISO 14971 topics given in Section IV. Space constraints do not permit us to address each topic in detail; we emphasize the issues that we believe are most important for the community to address at this point. Unless otherwise indicated, the use of the term "clauses" refers to clauses in ISO 14971.

A. Scoping, Boundaries, and Division of Responsibilities

In this section, we address issues associated with the overall scoping and allocation of responsibilities for achieving safety. In 14971, these issues are dispersed throughout the standard, but are dealt with most directly in the initial sections of Risk Analysis and in sections addressing labeling and instructions for use.

Clause 4.2 asks the manufacturer to describe the intended use/purpose, characteristics related to the safety, as well as reasonably foreseeable misuse. This information sets the

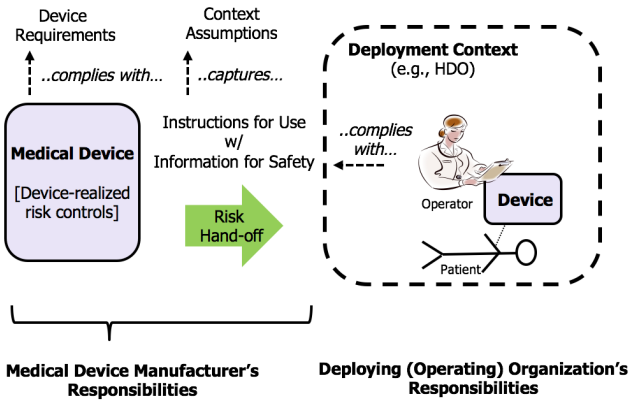

Fig. 4. ISO 14971 Implicit Risk-related Responsibilities

scope for the range of risks concerns to be considered, and forms a basis for identifying relevant notions of harm and hazardous situations.

Note the implied separation of responsibilities – while the manufacturer is responsible for identifying and preventing foreseeable misuse, exhaustive analysis and control of hazardous situations associated with "off label" use, which is not the manufacturer's responsibility. When deploying/applying the item into a care-giving context, the HDO is responsible for ensuring that their use of the device falls within the scope indicated by the intended use. Other divisions of responsibilities are reflected in ISO 14971's notion of *information for safety* (see Clause 6.2 and Annex J) – specifically, device labeling and instructions for use often indicate the actions that operators and maintainers must take to ensure that the device operates safely and effectively. Such actions may include device calibration (e.g., ensuring that pump fluid flow rates match pump specifications), proper physiological interfacing of device with the patient (e.g., ensuring that a pulse oximeter sensor clip is properly positioned to obtain accurate readings, ensuring that a PCA pump is appropriately primed to avoid infusion of air bubbles), and ensuring that device input settings are based on well-justified care-giving procedures and are carried out by persons with appropriate authorization.

An abstract summary of implied responsibilities (see Figure 4 is that the manufacturer is responsible to describing the function/purpose of the device, for capturing the medical function and risk controls in requirements, for verifying that the device realization complies with the requirements, and for providing clear and complete instructions for use (14971 Table E.1 indicates that incomplete labeling / instructions is a potential hazard), and for assuring that device will achieve its stated purpose with acceptable levels of risk provided that the HDO follows the instructions for use. The HDO, in turn, is responsible for complying with the instructions for use. Thus, the overall achievement of risk controls is divided between the manufacturer and the HDO and can be understood using the idiom of "contracts": The manufacturer is responsible for 1(a) clearly stating their *assumptions* about how the device will be operated and 1(b) developing and assuring that the device operates safely as long as the stated *assumptions are met*, while (2) the HDO is responsible for following the instructions for use.

This view of scoping in terms of purpose/function, product boundaries, allocation of responsibilities, and contractual relationships is *implied* by current risk management standards and guidance, but these notions *must be addressed more explicitly and comprehensively for interoperable systems*.

Challenges: Specification of Boundaries and Scope of Risk Management

First, to better specify allocation of responsibilities, item boundaries need to be specified with greater precision. We use the term "item boundary" to mean the point(s) at which the item interacts with its context; functionality inside the boundary is the responsibility of the item manufacturer, while entities and actions outside the boundary are not under the control of the manufacturer, and can only be indirectly controlled through item instructions for use. As healthcare interoperability technologies mature, interoperable system elements will likely become more fine-grained (e.g., making use of increasingly smaller and interchangeable sensing and actuation components). Moreover, architectures will become more flexible (i.e., more variability within architectures) as solutions aim to achieve broader applicability and/or greater customizability and tailoring to individual patient needs. Boundaries of the interoperable items may not fall along conventional medical device boundaries. For example, sensing and actuation items may increasingly have no operator interfaces or even forego incorporation of signal processing or control algorithms as those capabilities are "off-loaded" to general-purpose platform user interfaces and software apps executing on platforms. Infrastructure items such as the ICE Supervisor, Network Controller, and Data Logger do not have a conventional medical intended use, but may provide risk controls (e.g., authentication of devices, detection and notification of communication failures), and their reliability may be necessary for the safety of an interoperable system that they support. While many medical devices previously had limited and easily recognizable information/control pathways (e.g., the conventional operator interface), more modern devices may have multiple interaction points, including both wired and wireless network interfaces with wide-ranging capabilities exposed over those interfaces, as well swappable media (e.g., memory cards) – all of which must be explicitly indicated as within the scope of item risk management.[1] In the case where the item is itself an interoperable system (composed of sub-items whose combined functionality and interactions give rise to behavior whose safety and security must be assured), specifying the system boundary (and thus the scope of the system manufacturer's risk management) is also extremely challenging. In the ICE architecture, a system is usually considered to be constituted in terms of one or more apps, which together achieve a specific medical

[1]This also supports a characterization of the attack surface of an item which must be addressed in security aspects of risk management.

intended use, running on the Supervisor, interacting with one or more devices, with communication provided by the Network Controller. In such a case, while it is reasonable to assume that the local area network associated with the network controller is within the system's manufacturer's risk management scope (possibly reusing the assurance and risk management artifacts provided by the platform provider), it seems reasonable that the broader HDO IT network to which the ICE-based system may be connected resides outside the scope of the system manufacturer's risk management responsibilities – the system manufacturer can only state assumptions about how the IT network will be operated and protected. Even determining the scope of assumptions (i.e., determining the scope of the item's context to be addressed via labeling) is challenging – how much instructions/labeling are necessary and how far out into the IT network and HDO organizations should the assumptions reach? Moreover, when the system allows variability in the sub-items (e.g., in our example application, the system may have variations in which different pulse oximeters are used different executions of the system), declaration of risk management scope needs to indicate all the different variants to be addressed in the system risk management argument.

Possible solution concepts: The opening activities of ISO 14971 Section 4 Risk Analysis should be augmented to more precisely indicate the scope of risk management. A crucial element of this is identifying (at least at an abstract level), the boundary of the item in terms of interaction point(s) with its context. The boundary identification should be traceable to architecture and interface specifications produced in item design activities, and this in turn should be traceable to the item's realization. The boundary specification should be used to differentiate the functionality under the manufacturer's direct control from behaviors outside of the manufacturer's control that can only be indirectly controlled via Clause 6.2 information for safety (captured in labeling and instructions for use). A justification for this increased emphasis on the notion of boundary can be found in the foundational work on dependability concepts by Avizienis et al. [7] who stress importance of specifying the system boundary in terms of the *services* that the system provides to its context through defined interfaces. In situations where the item is a system (includes cooperating sub-items), the scoping should a description of the interoperable sub-items and the variability of the sub-items that will be addressed in risk management. While this will inevitably involve including more technical and engineering issues within the scope (i.e., there is a need for references to architecture and interface descriptions), approaches should be developed to enable an abstract and "light-weight" description of boundaries and variabilities that are refined in later development activities to more detailed architecture descriptions traceable to the initial boundary description.

Challenges: Expanding the Notion of Intended Use and Contexts of Use The purpose of 14971 Clause 4.2 includes indicating the care-giving function of the item from which top-level harms and hazardous situations are derived (e.g., the infusion purpose of the PCA pump leads to identification of hazards associated with over-infusion, under-infusion, and air embolisms). However, many interoperable items such as a Supervisor or Network Controller may not serve a distinct medical function. Instead, they have technical functions which enable and support medical functions. For such items, the top-level notions of patient harm and precise hazardous situations cannot be stated. However, the technical objectives including both functional goals, performance goals, reliability goals, and notions of risk controls provided – all of which will be relied on by application-level functionality – can be stated. For items that are interoperable medical devices or software (apps), the intentions of the item may be to interoperate with one or more platforms, upon which the device/app depends for its correct/safe behavior. This too is not a medical purpose, but a technical purpose that delimits the use of the device/app. A device may have many possible uses in different interoperable systems that are not known a priori by the manufacturer. However, contraindications may be stated (for example, a Bluetooth enabled consumer-oriented pulse oximeter may support health tracking applications, but may be indicated as unsuitable for clinical applications where a high level of precision is required from the telemetry). Thus, even the delimiting of medical function becomes more challenging. For the purpose of understand how faults originate and how they should be controlled, Avizienis et al. [7, Section 3.1] advocate for a clear distinction between the *development* and *use* phases / environments of an item. In the case of an interoperable item, in addition to its deployment and use in care-giving, the item may be "used" by *other development organizations* as they integrate the item into a platform asset base or system. This "development context of use" also needs to be guided by information for safety, e.g., to constrain how the system context should appropriately call the services of the provided item through its interfaces and how notifications from the item should be used to implement system level risk controls. As an example of the later case, an ICE network controller may notify the ICE Supervisor that a device upon which an executing app depends is suffering from a connection failure, and the app needs to realize a risk control by moving the system to a safe state.

Possible solution concepts: The scope of Clause 4.2 should be augmented to require both a medical *and* technical purpose/scope. For items such as medical devices, the medical purpose should be expanded to address the scope and contraindications in interoperable system contexts. Section 4 should also be augmented to require the manufacturer to specify the item's context (e.g., users, workflows, and external systems with which the item iteracts) to be addressed through information for safety. This should include both the *development context of use* (the nature of the architectures and system contexts into which the item is designed to be integrated) as well as the *deployment context of use* (the nature of the care-giving and clinical workflows that the item is designed to support). The development context of use should indicate the platforms or interoperability frameworks that the item

supports. These usage context descriptions should be also be used in subsequent risk management activities to state the competencies necessary to achieve development context and deployment context controls upon which the safety/security arguments for the item will depend.

B. Risk Analysis

Risk analysis includes identifying and documenting reasonably foreseeable sequences or combinations of events that can result in a hazardous situation. This includes identifying relevant harms and hazardous situations, and then analyzing how the effects of root causes, such as component failures, may propagate through the system and trigger hazardous control actions.

Conventional hazard analysis reports often only document the observable effects at the boundary of the component where the root cause occurs and then at the system boundary where a direct link to a notion of harm can be made (see for example [11, p. 249]. The propagation pathways and causality relationships between the originating component and the system boundary are often omitted or not described in detail. While this approach may be sufficient in monolithic single vendor systems, we believe that carrying out appropriate hazard analysis in multi-vendor distributed systems requires a different approach.

In the example system of Section III, a sensor degradation or failure in the pulse oximeter (PO) may give rise to values observed by clients on the pulse oximeter interface that are significantly higher or lower than the patient's true SpO2 status. Such a network-reported reading may lead the app to output a control signal (input to the PCA Pump via its network interface) that causes the pump to continue to infuse because the app control algorithm mistakenly determines that the patient is healthy enough to receive continued infusion. The continued pump infusion represents a hazardous control action (an observable effect at the system boundary) – the pump continues to infuse when it shouldn't – leading to possible harm in the form of an opioid overdose. We discuss various aspects of this example when describing the challenges below.

Challenge: Conventional terms such as harm/hazard/hazardous situation do not adequately address hierarchical system structures and technical context of use. As discussed in Section V-A, when the pulse oximeter manufacturer performs risk analysis, they do not a priori know all the interoperable systems in which the device may be used. Therefore, they would not identify the harm of over-infusion as a relevant patient harm their risk management process. The only relevant concern that they would identify is that the device must provide its SpO2 reading according to some technical specification over its interoperability interface.

In an even greater departure from convention, the Network Controller does not even have a direct care-giving function (its primary function is to move data from one component to another – it is agnostic regarding the clinical purpose of the data). Thus, specific patient harms that would drive risk management when following ISO 14971 Section 4 cannot be identified. Yet, the Network Controller does have relevant risk concerns – not directly related to patient harm, but to possible failure to transport data according to specification (e.g., to achieve certain declared quality of service and integrity properties).

Solution direction: Generalized vocabulary that accounts for hierarchical system structures and technical context of use. Conventional concepts of *harm*, *hazard*, and *hazardous situation* which are linked directly to patient safety in ISO 14971 need to be generalized to *risk concerns* that include meeting technical specifications that if violated have the possibility of causing a failure of a medical function for surrounding interoperable system context. IEC 80001 (see Section A.3) already makes a similar generalization when it uses the term *top-level property* to include effectiveness and security concerns (in addition to patient harm) and it modifies the ISO 14971 definition of *harm* to address reduction in effectiveness and breach of security.

Challenge: Imprecision and ambiguity in describing root causes and observable effects of root causes. In the example hazardous situation above, when reasoning about possible system-level patient harms, the app manufacturer would need to have an understanding of root causes and observable effects of faults originating in components coming from other manufacturers in their risk management files/reports. In addition to situations like the pulse oximeter failure above, the system-level safety reasoning will need to consider the effects of a network controller failure (dropped messages, late messages, corrupted messages) or supervisor failures (inadequate processor or memory resources to support the app's execution requirements), and failures of the PCA pump which might prevent the pump from appropriately responding to a control command to pause infusion. Different manufacturers may have different ways for documenting faults and for describing the visible effects on their item's interoperability interfaces. It is very difficult for an integrating item (such as the app in our example scenario) to perform effective risk analysis when it does not have a clear understanding of how errors may be manifested on the interfaces of its client items.

Solution direction: Nomenclature for faults and errors. The medical device community has a history of dealing with potential ambiguities in the semantics of important domain information by using taxonomies and nomenclature. For example, ISO/IEEE 11073-10101 provides and extensive nomenclature (syntax and informal semantics) for many dimensions of health informatics. A possible solution direction is to have (even a much smaller scale than 11073) a community effort to construct a nomenclature for interoperability-related failures and associated error effects that appear on component interfaces. Possible directions here include considering the extensive fault taxonomy of Avizienis et al. [7] and the Error Type Library from the AADL Error Modeling Language [36]. Initial proposals for core interoperability error nomenclatures have been given by Procter et al. [33], [34]. This type of approach is being

considered in the AAMI/UL 2800 interoperability safety and security committee work. In the context of platform-based development, even in the absence of standard nomenclatures, platform manufacturers or associated consortia can develop shared approaches for denoting faults/errors to be used across all platform stakeholders.

Challenge: Reporting of component error propagations. Once nomenclature/taxonomy approaches for denoting faults/errors are in place, there is a challenge of how to present that information in a meaningful way to stakeholders outside a manufacturing organizing. Interoperability item integration is typically concerned with the correct use of interoperability interfaces including required data representation, correct invocation procedures, timing properties, error return codes, etc. We believe it is natural to extend interface documentation to capture (via appropriate nomenclature) the types of errors (observable effects of failures) that may propagate *out* of a component. Corresponding, an item manufacturer may indicate that types of *inward* propagating errors that it addresses in risk management activities for its interoperability interfaces. With this type of document in place, item integration may more effectively recognize and resolve mismatches in risk management assumptions. For example, an integrator may be able to determine that an outward propagating error behavior for a data source component is not handled by the risk management activities of a manufacturer that is consuming that information. Building on this idea, item risk management documentation needs to capture intra-component error propagation information for flows and transformations of errors between a item's input and output interfaces. For example, in our error scenario above, the pulse oximeter manufacturer should document that their item is a possible source of a *value error* due to improper calibration. The app manufacturer should document that that *value errors* received from monitoring devices may cause it to generate *erroneous control signals* (leaving the pump to continue to infuse when it should be paused) as output to the pump, and the pump manufacturer should document that *erroneous control signals* (in particular, signals derived from decisions taken on an erroneous understanding of the patient's health) may lead to the pump continuing to infuse when it should be paused. It is important to understand that it is exactly this type of item-by-item causality information that is needed to perform, e.g., a system-level FEMA (uncovering that a failing pulse oximeter may report an erroneous value leading to an over-infusion harm). Item manufacturers can potentially make this abstract error flow information available to item integrators without disclosing proprietary details of the item implementations. Third-party certification can attest to the accuracy of the reported error flows.

Solution Direction: Capturing analysis results at item boundaries/interfaces using nomenclatures. Standard development activities should encourage manufacturers of interoperable items to extend interface documentation to include types of errors addressed by item risk management as well as error flow propagation paths through components. This type of information seems crucial to be able to develop a compositional approach to hazard/causality analysis. Forward-looking efforts should identify model-based safety/security analysis approaches to documenting error propagation information in item architecture and interface models. The Error Modeling Annex provided by AADL is a good illustration of this approach [36]. Platform manufacturers should include developer guidelines for documenting the results of hazard analysis at item boundaries.

Challenge: Avoiding separate approaches for reasoning about safety and security root causes. Security considerations are a crucial aspect of risk management for interoperable products. However, the community is still converging on an appropriate approach for integrating safety and security risk management. Risk analysis for safety and security appears to share many common elements. However, recent publications, e.g., as illustrated by AAMI TIR 57 *Principles for medical device security - Risk management*, suggest the possibility of separate but parallel cross-communicating processes. As more experience is gained with assuring interoperable devices, the community needs to explore how safety and security risk analysis can be more tightly integrated so as to avoid duplication of enabling analyses and to more quickly resolve trade-offs between security and safety controls.

Solution direction: Using data and control relations as the basis of analyses. Procter et al. [34] make the observation that the *observable* effects of security attacks on interoperability networks are often indistinguishable from network failures in real time.[2] For instance, a maliciously modified message may be indistinguishable from a message corrupted due to a network failure, late message arrivals due to failure of a network to prioritize traffic may be indistinguishable from messages delayed due to denial/degradation of service attacks. The authors suggest driving failure and causality analysis of interoperable networks with a common set of "guidewords" that encompass both failure-related safety and security concerns.

Challenge: Different levels of reliability and trustworthiness. Since one motivation for interoperability is to enable a varied combination of items, it is easy to fall prey to situations where an item I may depend on other items that do not have appropriate levels of trustworthiness, reliability, or security to support I's safety objectives. For example, a possible reworking of our example scenario to replace a higher reliability clinical pulse oximeter with a low cost consumer oriented pulse oximeter, or to replace the local network control with cloud-based storage of current state and application rules, or to use a Supervisor execution context that does not properly sandbox apps to keep them from interfering with one another should trigger concerns in the risk management process. These problems are exacerbated by business pressures to use off-the-shelf consumer platforms such as mobile phones, networks, and application execution and data storage facilities to support medical applications. While concepts of "safety class" exist in ISO 62304, such notions are not an integral part of ISO 14971 and IEC 80001.

[2] A post-hoc audit, with sufficient data, should be able to differentiate.

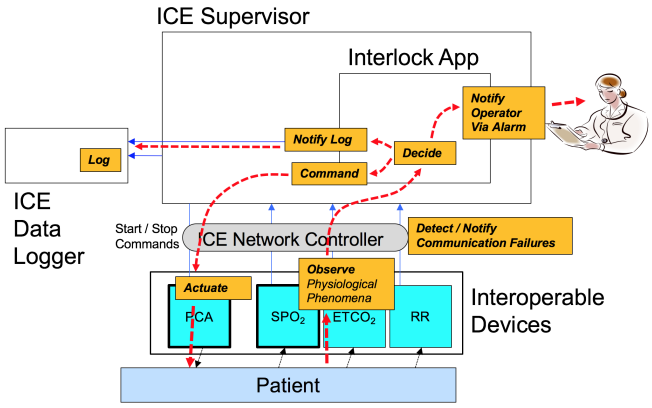

Fig. 5. Distributed Risk Controls

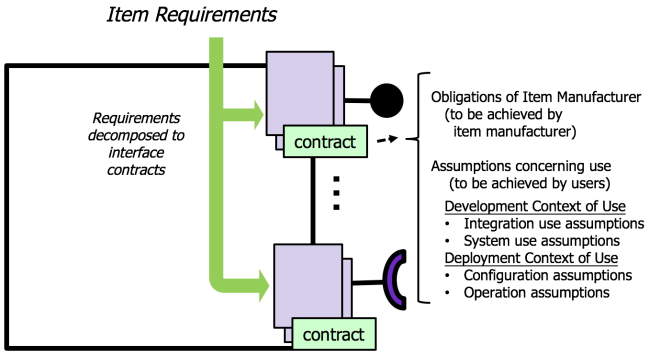

Fig. 6. Interface contracts capture assumptions on how interface is used and obligations that Item manufacturer must meet

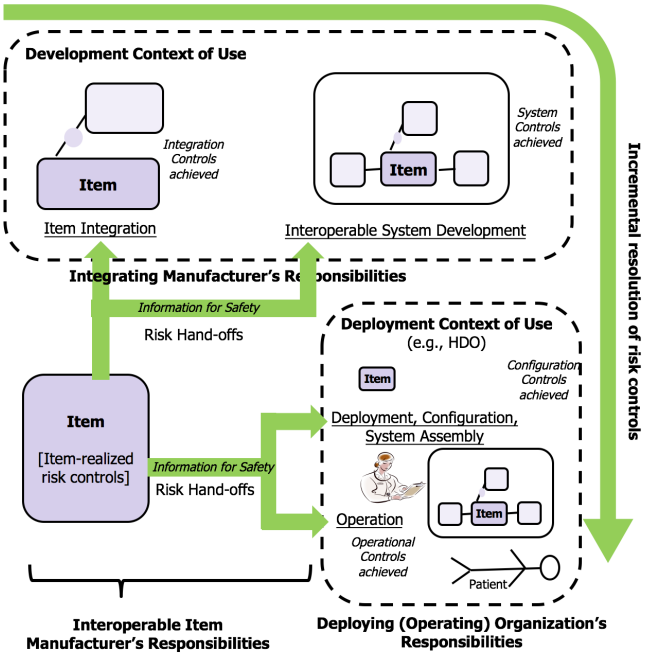

Fig. 7. Item allocation of risk controls to development context of use and deployment context of use

In contrast, functional safety standards such as IEC 61508 and its instantiation in IEC 26262 in the automotive domain deeply integrate the safety management process with hazard analysis and dependence analysis for determining safety integrity levels and provide stratified safety requirements for achieving differing degrees of item assurance.

Solution direction: Using data and control relations as the basis of analyses. Emerging medical device interoperability standards should begin to integrate notions related to safety integrity levels directly in risk management content. Risk analysis should include dependence analysis to ensure that an item I only has dependences on other items that are appropriate for I's safety objectives and risk controls. Following standards requirements should lead to manufacturers using arguments and objective evidence to justify claims of integrity levels. Platform manufacturers can play an important role by designing platform infrastructure with appropriate high-integrity solutions.

C. Risk Controls

Challenge: Risk controls are likely to be distributed across multiple items and responsible organizations: Realization of risk controls is often spread across a system architecture. It stands to reason that when a system is constructed by integrating multiple interoperable items, then control realization may be distributed across multiple items and multiple organizations. Therefore, better processes and design notations are needed to (a) understand design of such controls, (b) support decomposition and allocation of control elements items/manufacturers, (c) design general purpose risk control services in reusable infrastructure, and to (d) specify each item's assumptions about how other (externally realized) parts of risk control are to behavior and the obligations that an item has for ensuring that it's portion of the risk control functions correctly with appropriate reliability.

The example of Section III can be understood as implementing a risk control for a PCA over-infusion hazard, with different tasks for the control distributed across the subitems. Figure 5 illustrates that the pulse oximeter is responsible for *observing a phenomenon that may be indicative of the onset of a harm* (deteriorating respiratory health) and reporting the observation to the system context. The app is responsible for *deciding* if the sensor reading is indicative of deteriorating health, for *initiating operator notifications (alarms)*, for *logging* the problematic event, and for *initiating the actuation command* to pump to achieve a patient safe state. The pump is responsible for acting a command through its interface to actually *actuate* the patient to a safe state by halting infusion. Additional platform items support the control above by providing *operator display and notification services* (in the Supervisor) and *logging services*.

Solution directions: Recognizing the concept of distributed risk controls in emerging interoperability standards: ISO 14971 Clause 6 should be augmented to explicitly address the concept of distributed risk controls by including requirements that a manufacturer specify their item's obligations for one or more parts of a distributed risk control and that assumptions that the item makes about how the integrating

context will correctly use the control elements are explicitly captured by the manufacturer and externally communicated in the ISO 14971 concept of *information for safety* which is captured in instructions for use. Requirements should also ask manufacturers to specify how their parts of a risk control are manifested in the item's interoperability interfaces (see Figure 6).

Figure 7 provides a schematic illustration of how the simple notions of implied obligations of ISO 14971 in Figure 4 need to be enhanced to address multivendor interoperability. As an example instantiation of the schema, the PCA pump manufacturer will need to disclose to the integrator how the pump's interoperability interface is to be used and what controls need to be present to handle failure modes reflected on the pump's interface. The pump manufacturer needs to disclose to the system developer (who may be the same organization as the integrator) what patient conditions should typically lead to instructions being sent to pause pump infusion. These disclosures inform the proper engineering of controls for the pump's usage in the development context of use (situations where other developing organizations are using the pump in larger contexts). The pump manufacturer needs to disclose to deploying/operating organizations the appropriate methods for calibrating the pump, confirming the interoperability mechanisms are working correctly in the deployed context, configuring the access control policies of the pump to align with the authorization policies of the using organization, and instructions for using the pump (and its interoperabiity features at the point-of-care). These disclosures inform the deployment context of use of appropriate controls. In summary, overall risk is controlled by (a) the pump manufacturer assuring that device meets its requirements (including interface contracts), (b) the pump manufacturer appropriately specifying information for safety regarding the pump use in both system development and operation, (c) integration and system development actitivies correctly following the pump's instructions for development use, and (d) deploying/operating organizations correctly following the pump's instructions for deployment/operational use. The arrow wrapping around the top and right sides of the figure indicates that each of these steps can be seen as achieving a progression of risk controls across different organizations, with each activity further contraining/controling the pump – leading to an acceptable level of risk for the pump as it is being used in patient care.

Solution directions: Modeling technologies and contract-based approaches for designing and specifying distributed risk controls Modeling notations should be developed to show decomposition of risk controls into basic elements, perhaps using the system-theoretic idioms of *observing phenomena*, *deciding*, *notifying*, *commanding*, and *actuating*, etc. outlined above. Contract idioms [15], [27] should be considered to explicitly state the assumptions and obligations for an item's portion of a risk control.

Challenge: Rapid and consistent development of reliable risk controls: App-based interoperability encourages rapid development and innovation, often by organizations that may not be familiar with safety engineering. In this context, there is a danger of inappropriately designed controls. For example, developers may focus on software app development while failing to address end-to-end information and control pathways that factor through hardware sensors/actuators and middleware – each with their own distinct failure profiles, latencies, and accuracy specifications. App developers may fail to take into account variabilities that result when apps are executed with different versions of platforms for sensing/actuation components.

Another common pitfall is failing to identify unsafe emergent properties that arise when interoperable items are composed. For example, an app manufacturer may not anticipate safety issues that arise when their app is executed on a platform with other apps running simultaneously. In such cases, other (perhaps non-critical) apps may interfere with the execution of a critical app by over-utilizing memory, processing, or communication resources. In addition, an app may seek to maliciously interfere with another app.

Finally, the need to appropriately verify and establish reliability of controls may fail prey to business pressures associated with moving product to market.

Solution direction: Design of high-assurance safety services in platforms: Platform-based solutions over excellent opportunity to provide well-designed, high-assurance safety services that can be easily reused by manufacturers using the platform to develop interoperable systems. The ISOSCELES Platform [9] is an example where hypervisor technology is being applied to "sandbox" different modules of a medical device so that they do not interact with each other than through explicitly declared communication channels (which can be subjected to focused scrutiny during assurance processes).

D. Risk Control Verification and Evaluation

Challenge: Verifying individual risk control elements: In the interoperability context, verification of risk controls will often be distributed across the activities of item development, integration, and system development. Since item manufacturers may only be implementing individual elements of risk controls as discussed above, they will need to test their elements without full knowledge of the specific system context into which their items will be integrated. Testing against representative system contexts will play a valuable role, but the ultimate objective will be to assure that correctness and reliability of the risk control element under any system context that satisfies assumptions captured in the contract-based specification associated with the element. Item integration activities need to test that contract assumptions of item control elements are satisfied and should confirm via testing that item's obligations for the risk control element are met. System development needs to perform end-to-end testing that risk controls are correct and effective for controlling system-level hazardous situations.

Solution directions: Recognizing the concept of distributed risk controls in emerging interoperability standards: Emerging safety/security standards should augment ISO 14971

Clause 6.4 to explicitly address the verification responsibilities outlined above for item development, item integration, and system development. A central concept in the augmentation should be the orientation of test design around the notion of *contract* (see Figure 6 that explicitly captures item obligations and context assumptions related to the behavior of the item as exposed on interfaces (see [27] for foundations and initial directions).

Solution directions: Reusable platform-based testing infrastructure: To achieve the above objective of "testing in representative system contexts", interoperability frameworks should provide test beds that include simulated devices and representative applications, as well as fault injection capabilities that simulate communication failures and other interoperability-related issues. Conditions for conformity to interoperability frameworks should be established based in part on testing criteria phrased in terms of the platform test bed.

Challenge: Developing appropriate notions of test coverage for interoperability variations: Interoperable products are designed to be reused in different contexts (the external contexts of a product can vary) and in different configurations (the internal components, e.g., of a system, can vary when one interoperable product is replaced with another that provides similar functionality). One of the primary purposes of a platform is to support variability – the devices, apps, and services integrated to the platform are changed for different applications. While this flexibility has many benefits, it poses many challenges to risk management verification. An interoperable device needs to tested with a sufficiently rich context behaviors to ensure that it can perform as specified when placed into an arbitrary context. A system that can support the swapping of devices needs to perform safely no matter what device set is chosen to instantiate the system. A platform needs to provide its services correctly under a potentially unbounded number of combinations of devices and apps. Even greater challenges arise with plug-and-play devices where a new unanticipated device can at run-time communicate its capabilities and the system (perhaps supported by the platform) must check to see if those capabilities are suitable for the system's objectives. Traditional risk management concepts must be augmented to address these different notions of variability, and risk control verification in particular must be enhanced to ensure that possible variations of an item have been verified to extent that enables one to have confidence in the safety of instantiations of the variability.

Solution directions: Explicit declaration of the scope of variabilities: Scoping activities of risk management (as in ISO 14971 Clause 4 and in design of risk controls as in ISO 14971 Clause 6)) must be augmented to explicitly declare the variation possibilities of a device/system/platform and the range of variations that will be considered in the risk management argument. Some of the most straightforward approaches are analogous to white-listing – e.g., the variability description of system might simply enumerate all the different models of medical devices that could be used to create system instances. But even this solution is challenging when a new medical device compatible with system needs is produced – the variability tracking of all deployments the system may need to be updated to reflect the authorization of the new device. Moreover, any realistic approach that allows multiple apps to run simultaneously with multiple devices would need to analyze combinations of apps or devices for unsafe interferences. The combinatorial explosion associated with this would stress white-listing approaches (e.g., one must move from white-listing individual items to combinations of items).

Solution directions: New notions of coverage for variability: We argued above that interoperable item verification needs to utilize a "sufficiently rich" test suite to exercise the variabilities. The community is not now in a position to characterize "sufficiently rich". New foundational notions of *test coverage* are needed to be able to characterize the degree to which a test suite associated with an interoperable item provides coverage of its variabilities.

E. Risk Management Report

ISO 14971 Annex A.2.8 indicates that the risk management report is a "crucial part of the risk management file", and that "it is intended to be a summary of the review of the final results of the risk management process". In addition, the report "provides evidence that the manufacturer has ensured that the risk management plan has been satisfactorily fulfilled." This explanation seems to indicate that the risk management report provides an entry point to understanding the overall outcomes of the risk management activity and could aid in a "drill down" into the details and "evidence" provided by the risk management file.

Challenge: Phrasing of partial analysis results and partial risk controls in such a way that these can be easily understood and consumed by integration activities: The language in ISO 14971 Clause 8 suggests that the supporting conformity assessment is the primary purpose of the risk management report. However, in the context of distributed development, stakeholders other than conformity assessment bodies may also need "entry points" into the details of risk management information. Based the presentations in Sections V-B and V-C, one would expect such information to be included in disclosures from item manufacturers to both development users (e.g., integrators of items, users of platforms) and operational users (e.g., HDOs). Issues of primary importance here would be how to summarize item-level risk analysis, risk controls, and risk control verification which from a system perspective may only be "partial" (e.g., results of item-level risk management need to be integrated to establish system-level risk management). Moreover, it is important that such high-level report information be organized to phrase results in terms of the services provided on item interfaces, because integration activities are inherently interface focused.

Solution directions: Notations for arguments and evidence (e.g., assurance cases linked to architecture model-based capture of important risk information: ISO 14971 A.2.8

uses the words "high-level summary" and "evidence". It is possible to understand these in terms of the notion of assurance case which provides high-level arguments and claims that link to evidence (see ISO/IEC 15026-2). This suggests that an assurance case approach might be help in describing the what an item is achieving in risk management versus what will be addressed by the item context (i.e., the system context). Since the reporting will benefit from organization in terms of interfaces, phrasing results either in terms of or with links to architecture descriptions of interfaces may also be useful.

Challenge: Communication with external stakeholders and balancing the need to release risk management details for appropriate integration and use while avoiding the release of propriety information: An overarching challenge of the disclosure concept is for an item manufacturer is to balance (a) the need to provide enough information externally to enable development users to appropriately integrate the item into large contexts while (b) protecting intellection property and other proprietary issues. It is likely that targeted non-disclosure agreements between item manufacturers and development users will play a large role until the community develops a better understanding of how to handle this issue. As noted earlier, third-party certification can also play a key role in helping ensure trust.

VI. Conclusion

In this paper we have attempted to develop a consolidated description of some of the unique challenges in risk management for interoperable medical systems and Medical Application Platforms. Risk management in this context is particularly difficult since the process is "distributed" – not handled by a single entity such as a device manufacturer, but instead must be addressed via cooperation between operators, component manufacturers, and system integrators. The increased variability of integration environments also a significant challenge in reasoning about extant and *potential* risk, which may emerge if the operating environment changes.

We based our discussions and organized potential solution directions on ISO 14971, the primary medical device risk management standard. In particular, we have suggested how the standard's current focus on *stand-alone systems* can be augmented by emerging interoperability safety and security standards to better address *integrated* systems of systems, i.e., those composed of multiple devices, platform infrastructure, and software applications.

A. Ongoing and Future Work

In our ongoing work, we are developing a comprehensive set of objectives for distributed risk management that can be used to write assurance cases for devices and interoperable systems using medical application platforms. These would be usable by device manufacturers, system integrators, as well as platform developers. We are also developing automated tools for MAP risk analysis based on the Architecture and Analysis Definition Language (AADL) standard [38]. These tools emphasize formal annotations and rigorous approaches for architecture modeling and for capturing information flow and propagation paths of the effects of faults and errors through the architecture. The tools are designed to address many of the challenges discussed in this paper – particularly those related to risk analysis. Finally, we are continuing our work on open source medical device hardware and software infrastructure that provides architectural principles, partitioning technologies, and general purpose medical safety and security services that can enable interoperable device manufacturers to rapidly develop and assure risk controls [9].

References

[1] Center for Medical Interoperability. `http://medicalinteroperability.org/`, 2017.

[2] AAMI. TIR57: Principles for medical device information security risk management. Technical report, AAMI, 2016.

[3] 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.

[4] D. Arney, S. Weininger, S. Whitehead, and J. Goldman. Supporting medical device adverse event analysis in an interoperable clinical environment: Design of a data logging and playback system. *ICBO*, 833:335–339, January 2011.

[5] ASTM. F-2761: 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. Standard, 2009.

[6] ASTM Committee F-29, Anaesthetic and Respiratory Equipment, Subcommittee 21, Devices in the integrated clinical environment. Medical devices and medical systems — essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE), 2009.

[7] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.

[8] A. Besting, S. Bürger, M. Kasparick, B. Strathen, and F. Portheine. Software design and implementation concepts for an interoperable medical communication framework. *Journal of Biomedical Engineering / Biomedizinische Technik*, 2017.

[9] T. Carpenter, J. Hatcliff, and E. Y. Vasserman. A reference separation architecture for mixed-criticality medical and iot devices. In *Proceedings of the ACM Workshop on the Internet of Safe Things (SafeThings)*. ACM, November 2017.

[10] DocBox. ICE Demonstration Press Release. `http://www.docboxinc.com/press/20120209.asp`.

[11] C. A. Ericson II. *Hazard analysis techniques for system safety*. John Wiley & Sons, 2005.

[12] FDA. *Recognized Consensus Standards*.

[13] L. Feng, A. L. King, S. Chen, A. Ayoub, J. Park, N. Bezzo, O. Sokolsky, and I. Lee. A Safety Argument Strategy for PCA Closed-Loop Systems: A Preliminary Proposal. In V. Turau, M. Kwiatkowska, R. Mangharam, and C. Weyer, editors, *5th Workshop on Medical Cyber-Physical Systems*, volume 36 of *OpenAccess Series in Informatics (OASIcs)*, pages 94–99, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[14] J. Hatcliff, A. King, I. Lee, M. Robkin, E. Vasserman, A. MacDonald, S. Weininger, A. Fernando, and J. M. Goldman. Rationale and architecture principles for medical application platforms. In *ICCPS*, 2012.

[15] J. Hatcliff, G. T. Leavens, K. R. M. Leino, P. Müller, and M. Parkinson. Behavioral interface specification languages. *ACM Comput. Surv.*, 44(3):16:1–16:58, June 2012.

[16] J. Hatcliff, E. Y. Vasserman, S. Weininger, and J. Goldman. An overview of regulatory and trust issues for the integrated clinical environment. In *Joint Workshop On High Confidence Medical Devices, Software, and Systems & Medical Device Plug-and-Play Interoperability (HCMDSS/MD PnP)*, 2011.

[17] J. Hatcliff, A. Wassyng, T. Kelly, C. Comar, and P. L. Jones. Certifiably safe software-dependent systems: Challenges and directions. In *Proceedings of the on Future of Software Engineering (ICSE FOSE)*, pages 182–200, 2014.

[18] ICE Alliance. ICE alliance.

[19] IEEE. IEEE 11073-10207-2017 - IEEE Approved Draft Standard for Domain Information and Service Model for Service-Oriented Point-of-Care Medical Device Communication. Standard, 2017.

[20] M. Kasparick, F. Golatowski, and D. Timmermann. A safe and interoperable distributed alarm notification system for PoC medical devices using IEEE 11073 SDC. In *IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*, 2017.

[21] M. Kasparick, M. Rockstroh, S. Schlichting, F. Golatowski, and D. Timmermann. Mechanism for safe remote activation of networked surgical and PoC devices using dynamic assignable controls. In *EMBC*, 2016.

[22] M. Kasparick, M. Schmitz, B. Andersen, M. Rockstroh, S. Franke, S. Schlichting, F. Golatowski, and D. Timmermann. OR.NET: A service-oriented architecture for safe and dynamic medical device interoperability. *Journal of Biomedical Engineering / Biomedizinische Technik*, 2018.

[23] Y. J. Kim, S. Procter, J. Hatcliff, V.-P. Ranganath, and Robby. Ecosphere principles for medical application platforms. In *IEEE International Conference on Healthcare Informatics (ICHI)*, 2015.

[24] A. King, D. Arney, I. Lee, O. Sokolsky, J. Hatcliff, and S. Procter. Prototyping closed loop physiologic control with the medical device coordination framework. In *ICSE Companion*, 2010.

[25] A. King, S. Chen, and I. Lee. The MIDdleware Assurance Substrate: Enabling strong real-time guarantees in open systems with openflow. In *IEEE Computer Society symposium on Object/component/service-oriented realtime distributed computing (ISORC)*. IEEE, 2014.

[26] 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 *International Conference on Software Engineering (ICSE)*, pages 141–151, 2009.

[27] B. R. Larson, Y. Zhang, S. C. Barrett, J. Hatcliff, and P. L. Jones. Enabling safe interoperation by medical device virtual integration. *IEEE Design Test*, 32(5):74–88, Oct 2015.

[28] MDPnP Program. Openice – open-source integrated clinical environment, 2015. `https://www.openice.info/`.

[29] OR.NET e.V. OR.NET e.V. – safe, secure and dynamic networking in the OR, 2017.

[30] S. Procter and J. Hatcliff. An architecturally-integrated, systems-based hazard analysis for medical applications. In *2014 Twelfth ACM/IEEE Conference on Formal Methods and Models for Codesign (MEMOCODE)*, pages 124–133, Oct 2014.

[31] S. Procter and J. Hatcliff. An architecturally-integrated, systems-based hazard analysis for medical applications. In *MEMOCODE*, 2014.

[32] S. Procter, J. Hatcliff, and Robby. Towards an AADL-based definition of app architecture for medical application platforms. In *SEHC Workshop*, 2014.

[33] S. Procter, J. Hatcliff, S. Weininger, and A. Fernando. Error type refinement for assurance of families of platform-based systems. In *International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE)*, 2015.

[34] S. Procter, E. Y. Vasserman, and J. Hatcliff. SAFE and secure: Deeply integrating security in a new hazard analysis. In *SAW*, 2017.

[35] J. M. Rushby. Design and Verification of Secure Systems. *Operating Systems Review*, 15, 1981.

[36] SAE AS-2C Architecture Description Language Subcommittee. SAE Architecture Analysis and Design Language (AADL) Annex Volume 3: Annex E: Error Model Language. Technical report, SAE Aerospace, June 2014.

[37] C. Salazar and E. Y. Vasserman. Retrofitting communication security into a publish/subscribe middleware platform. In *SEHC Workshop*, 2014.

[38] Society of Automotive Engineers. Architecture Analysis & Design Language (AADL). Aerospace Standard AS5506, 2004.

[39] SurgiTAIX AG. SDCLib. `https://github.com/surgitaix/osclib/`.

[40] SurgiTAIX AG. SOFTice. `https://bitbucket.org/surgitaix/softice/`.

[41] UL. UL 2900-2-1 Software Cybersecurity for Network-Connectable Products, Part 2-1: Particular Requirements for Network Connectable Components of Healthcare and Wellness Systems. Standard, 2017. `https://standardscatalog.ul.com/standards/en/standard_2900-2-1_1`.

[42] E. Y. Vasserman and J. Hatcliff. Foundational security principles for medical application platforms. In *WISA*. 2014.