Ecosphere Principles for Medical Application Platforms

Yu Jin Kim Sam Procter John Hatcliff Venkatesh-Prasad Ranganath Robby
{yujin, samprocter, hatcliff, rvprasad, robby}@k-state.edu
CIS Department, Kansas State University, Manhattan, Kansas, U.S.A.

***Abstract*—A Medical Application Platform (MAP) enables multi-vendor heterogeneous medical devices to be integrated via network infrastructure and provides an application hosting facility that supports a wide range of clinical applications for data fusion, decision support, multi-device safety interlocks, workflow automations, and closed-loop control of actuating medical devices. The assurance of MAP components and systems is distributed across a broad group of stakeholders including medical device manufacturers, platform infrastructure providers, application vendors, third-party certification organizations, and regulatory agencies. Realization of the MAP vision requires that all stakeholders involved in developing, testing, certifying, regulating, purchasing, and using medical devices and applications operate and cooperate within a well-defined ecosphere. This paper presents a high-level overview of the organization of such an ecosphere. We focus on identifying stakeholder roles, responsibilities, artifacts, and interactions; we also indicate the contributions of each of these to the development and safety/security assurance of MAP-based systems.**

I. Introduction

The emerging notion of a Medical Application Platform (MAP) [4] brings a *system of systems* approach to providing interoperability of heterogeneous medical devices and Health Information Systems (HISs). Hatcliff et al. write that: "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 (i.e., *apps*) that provide medical utility via the ability to both acquire information and update/control integrated devices, IT systems, and displays." In MAP architectures, devices and IT systems are *service components*, which provide sensing, actuation, and information capabilities to *application components* (i.e., apps); apps run on *infrastructure components*, which provide real-time aware network communication of data and control between service and application components as well as operating system functions for hosting processes that execute apps. MAP apps can provide a range of medical utility (e.g., smart alarms, safety interlocks, decision support algorithms, etc.) [4]. Using an advanced interface description language, an app states the capabilities that it *requires* from service components, service components state the capabilities that they *provide*, and infrastructure components ensure that an app's required capabilities are satisfied by the service components' capabilities (otherwise, the app is not allowed to launch). MAPs are distinct from existing medical systems and other safety-critical cyber-physical systems because of their unique characteristics and the "ecosphere approach" required for their development, assurance, and deployment.

- **Integration of Heterogeneous Components:** MAP components may be produced by different vendors. Integration and systematic reuse of these heterogeneous components requires interoperability; vendors will need to comply with consensus-defined interfaces and implement specified functionalities.
- **Interchangeable Components:** In a MAP, a component from one vendor can be replaced by one or more components from other vendors as long as the substituted component(s) can provide the capabilities required by an app. Thus, an app vendor must be able to trust that the service component vendor has correctly disclosed the component's functional capabilities and operational states via its interface and that the component's true behavior is compliant with that interface.
- **Assembly at the Point-of-Care:** *Assembly* – the activity of physically plugging together components – occurs at the deployment site such as a hospital or other Healthcare Delivery Organization (HDO). Tasks associated with *integration* – the specification of system requirements and assurances that those requirements are satisfied when the app executes on the platform with appropriate service components – are carried out by the app vendor. The assurance obligations associated with establishing service and infrastructure component behavior compliance to disclosed specifications are distributed across the ecosphere.
- **Safety- and Security- Critical:** MAPs are safety-critical because their malfunction could cause serious injury or death. In addition, MAPs are security-critical because the information exchanged in a MAP may be sensitive health data, so unauthorized access could enable a malicious user to cause significant harm.
- **Assurance Reuse and Component-wise Safety Reviews:** MAPs are designed to support plug-and-play interoperability and interchangeable components. The paradigm rests on being able to bring to market assured components that can then be reused in a variety of combinations, including combinations which were not anticipated at the time that the components were developed. However conventional safety regimes, associated standards, and regulatory paradigms typically focus on establishing the safety of complete integrated systems (instead of safety-related properties of reusable components), and they often require a system to be completely re-certified whenever a single component is interchanged with another. New compositional approaches to safety and assurance must be developed that enable individual components to be certified to conform to interface specifications and then the resulting

This work was supported in part by the U.S. National Science Foundation (NSF) awards OCI-1239543, CNS-1238431, CNS-0932289, and by CIMIT/Massachusetts General Hospital as a subcontract of a NIH/NIBIB Quantum grant.

assurance reused to establish that assembled systems satisfy system-level safety and security properties [6].

The characteristics above imply that new engineering solutions (in the form of architectures and interfacing technologies), development and assurance processes, safety and security standards, and organizational paradigms are needed to ensure that MAP-based systems can be effectively developed and safely deployed. Architectural solutions for MAPs (e.g., ASTM 2761 [3]) are emerging, and standards for safety, security, and essential performance of interoperable medical systems (e.g., AAMI UL 2800) are in development. However, the community has not yet clearly identified the basic principles of appropriate organizational paradigms and associated processes that must accompany these.

In this paper, we argue that the concept of an explicitly recognized and organized ecosphere of stakeholders is necessary because no single entity bears the responsibility for design, development, integration, and assurance in MAP-based systems. Instead, these activities are distributed across stakeholders who each have their own potentially conflicting mission goals and operational tempos. Hence, we define an *interoperability ecosphere* as the collection of stakeholders that are involved, artifacts that are produced, processes that are followed, and trust relationships that are established, to develop, assure, market, deploy, and operate interoperable systems. The driving tenet being, for MAPs, the interoperability ecosphere must lead to medical systems safety and security. In short, the main contribution of this paper is the definition of the interoperability ecosphere for MAPs in terms of stakeholders, their responsibilities, and their dependences on other stakeholders in terms of tasks and artifacts.

The rest of the paper is organized as follows. Section II gives an overview of the Integrated Clinical Environment (ICE) architecture as defined in ASTM 2761 to illustrate the types of components and interfaces that are relevant to a MAP ecosphere. Sections III and IV summarize the ecosphere stakeholders, tasks, and representative processes by following the development of an example medical system. We conclude in Section V.

II. Background

MAPs can be realized via different architectures. The Integrated Clinical Environment (ICE) [3], standardized in the ASTM F2761, is one such architecture; ICE development has been led by the CIMIT Medical Device Plug-and-Play (MD PnP) interoperability project. ASTM F2761 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 [1].

Figure 1 illustrates the ICE architecture, where boxes comprised of dashed/dotted lines depict ICE components and thick dashed lines indicate the exposed interface of adjacent components. On connection, ICE-compatible equipment (i.e., *ICE Devices*) transfer the description of their capabilities and behaviors to the *ICE Platform*; this description is referred to as the device model (*ICE DM*). These capabilities are then used by ICE apps to provide medical utility. *ICE Devices* communicate with the *ICE Platform* through Interface 2 and

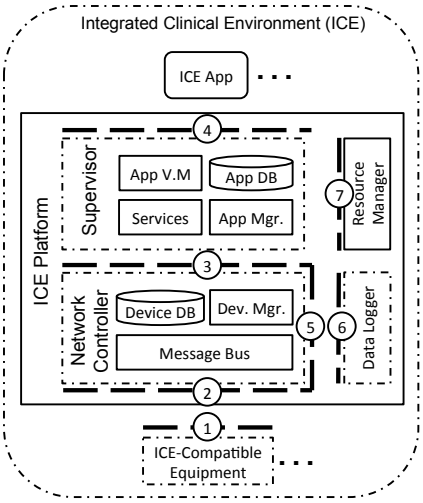

Fig. 1. ICE Architecture with MDCF components: ICE concepts are represented in dashed dotted lines and MDCF components are in solid lines.

ICE Devices are accessed through ICE equipment interfaces (*ICE EI*, Interface 1).

The *ICE Platform* consists of two major components: a *Network Controller* and a *Supervisor*. The *Network Controller* has two primary tasks: (1) to act as a communication hub between *ICE Devices* and *Apps*, and (2) to provide services for the *Supervisor* on behalf of the *ICE Devices*. The goal of these services is to provide the functional capabilities and performance guarantees (accessed via Interface 3) from *ICE Devices* that the *Supervisor* requests.

The *Supervisor* hosts medical apps that provide medical utility such as smart alarms, clinical decision support, safety interlock, etc. In addition, functional (e.g., SpO_2 measurement) and non-functional (e.g., measurement report latency) capabilities that *Apps* require are provided through the *Supervisor* interface (Interface 4), which constitutes, in essence, an Application Programming Interface (API) for the *Apps*.

The Medical Device Coordination Framework (MDCF) (e.g., [10]) is a prototype implementation of ICE jointly developed by researchers at Kansas State University and the University of Pennsylvania. Components added to the MDCF are presented in solid-lined boxes in Figure 1. The MDCF provides a middleware substrate and associated services [9], tools for authoring apps, generating executable APIs [13], [8], and performing risk management activities [12].

III. Stakeholders

This section describes the primary stakeholder categories in an interoperability ecosphere for MAPs, along with the *tasks* they perform, the *artifacts* they produce, and their *dependencies* on other stakeholders. To make the discussion concrete, we present ecosphere concepts in terms of the ICE architecture. A complete list of stakeholders' tasks will require further research and discussion, but in this paper, we provide some envisioned tasks that will help each stakeholder achieve their goals and, in the process, promote the viability and effectiveness of the ICE ecosphere. Identifiers assigned for each *task* and *artifact* are referred to as *x*T.*n* and *x*A.*n* in the rest of document, respectively, where *x* is a capital letter that represents the stakeholder and *n* is a number.

Figure 2 captures some tasks performed by stakeholders and the produced artifacts in the context of ICE Device development.

A. Stakeholder Consortium

Description: Interoperability in MAPs is enabled by a well-defined architecture that specifies component boundaries and interfaces as well as process requirements for regulating component compliance. Therefore, standardization of architecture and processes is crucial to the seamless integration of MAP components. To provide a central organizational authority for an interoperability ecosphere, the ecosphere will include a stakeholder consortium. Existing examples of interoperability consortia include the Continua Alliance, the WiFi Alliance, and the Industrial Internet Consortium. In the ICE context, the newly formed ICE Alliance will likely play the role of the consortium. In the ICE ecosphere, the consortium works with standards development organizations (SDOs) to develop a family of standards that ensure safe, secure, and effective interoperability/integration of components in an ICE system. Consortium members include stakeholders in the ICE ecosphere (e.g., device vendors, app developers, regulatory authorities) who provide requirements to advance the standards.

Tasks: The consortium organizes events and workgroups in order to elicit requirements from the various stakeholders in order to develop ICE standards. The consortium also provides tooling and supporting artifacts in order to facilitate development of ICE components. Specifically, the consortium performs the following tasks:

CT.1 Develop architecture standards **(CA.1)** for ICE systems (e.g., [3]) that define components and their functionality.
CT.2 Develop service interfaces for platform components (e.g., *Network Controller*, *Supervisor*, *Data Logger*) as part of the Platform Component Standard **(CA.2)**.
CT.3 Develop standards for ICE Device Modeling Language (*DML*) used to describe device capabilities **(CA.3)**. This standard also defines the translation scheme from *DML* to commonly used programming languages in order to enable interoperable low level communication (e.g., OMG IDL [2]).
CT.4 Provide for the collection of domain-specific data types (collectively referred to as the *modeling vocabulary*) and common classes of devices (e.g., pulse oximeters or infusion pumps; referred to as *reference device models*) in order to enable code reuse / efficient development **(CA.8)**.
CT.5 Develop standards for the ICE App scripting language that describes application behavior **(CA.4)**. The scripting language should specify, for example, the app's required resources (e.g., CPU, memory, sensor data stream(s)) and logic for processing in order to provide clinical utility.
CT.6 Develop risk management guidelines **(CA.6)**. Each ICE component contains unique safety concerns, so the guidelines provide information regarding assessment methodology and documentation (e.g., data requirements as evidence of component effectiveness) for regulatory review.
CT.7 Develop standards for verifying the compatibility between the provided capabilities of an *ICE Device* and the required capabilities of an *ICE App* **(CA.2)**. This might include, for example, compatibility criteria for $EtCO_2$ between various units (e.g., mmHg, kPa, %).
CT.8 Develop process standards to support ICE component compliance **(CA.5)**. This standard should illustrate, for example, testing and verification procedures for compliance between a component interface and the component's actual behavior. This standard also provides template documentation for the compliance certification application **(IA.1)** and the regulatory approval submission **(IA.2)**.
CT.9 Approve and distribute tools that support interface composition and compliance certification processes **(CA.7)**. Tooling related to the *ICE Device*, for example, should provide a development environment for authoring and testing device interfaces. It should also support the translation of *ICE DM* into executable APIs in (potentially) vendor-specific programming languages as well as the generation of documentation that complies with process standards **(CA.5)**. Tooling can assist the work of preparing a component for regulatory review in, e.g., hazard analysis [12] and assurance case construction [11].

Artifacts: As the result of above tasks (**CT.1**-**CT.9**), the consortium produces a family of *ICE standards* that includes **(CA.1)** Architecture Standards, **(CA.2)** Platform Component Standards, **(CA.3)** Device Standards, **(CA.4)** App Standards, **(CA.5)** Process Standards for Compliance and Regulatory Submission, and **(CA.6)** Risk Management Guidelines. The consortium also provides **(CA.7)** *tooling* for component authoring, testing, reviewing, etc., and **(CA.8)** reusable *supporting resources* like reference device models or documentation templates.

Dependencies: Stakeholder requirements and participation are the basis of the tasks and artifacts that the consortium produces. Requirements from each stakeholder are referred to as *x*A.0, where *x* denotes a stakeholder. The leftmost vertical swimlane in Figure 2 depicts all stakeholders that provide input (in the form of requirements) to the consortium.

B. ICE Component Vendors

Description: ICE Component Vendors create and produce ICE components such as ICE Apps, ICE Devices, and ICE Platform Components (e.g., the *Supervisor*, *Network Controller*, and *Data Logger* components). In order to build ICE compliant components, each vendor conforms to interface and architectural standards and follows the compliance certification processes that have been defined by the consortium. Specific component development procedures vary among vendors and the required process standards **(CA.5)** for various ICE component types differ. Due to lack of space in this paper, we do not discuss these differences in detail here. Details of the process, broken out by component type, are presented in [7].

Tasks: Regardless of vendor-specific practices, the following tasks must be achieved to produce ICE-compliant components:

IT.1 Compose interface specifications of components that describe the provided and required capabilities. Using consortium-provided tools, vendors generate executable APIs and implement them to expose their component's functionality. Vendors then generate test suites from interface specifications to perform preliminary compliance testing.
IT.2 Submit a *compliance test application package* **(IA.1)** to a third-party certification authority for compliance testing. *Compliance test application package*s consist of a release-ready version of a component and any documentation expected

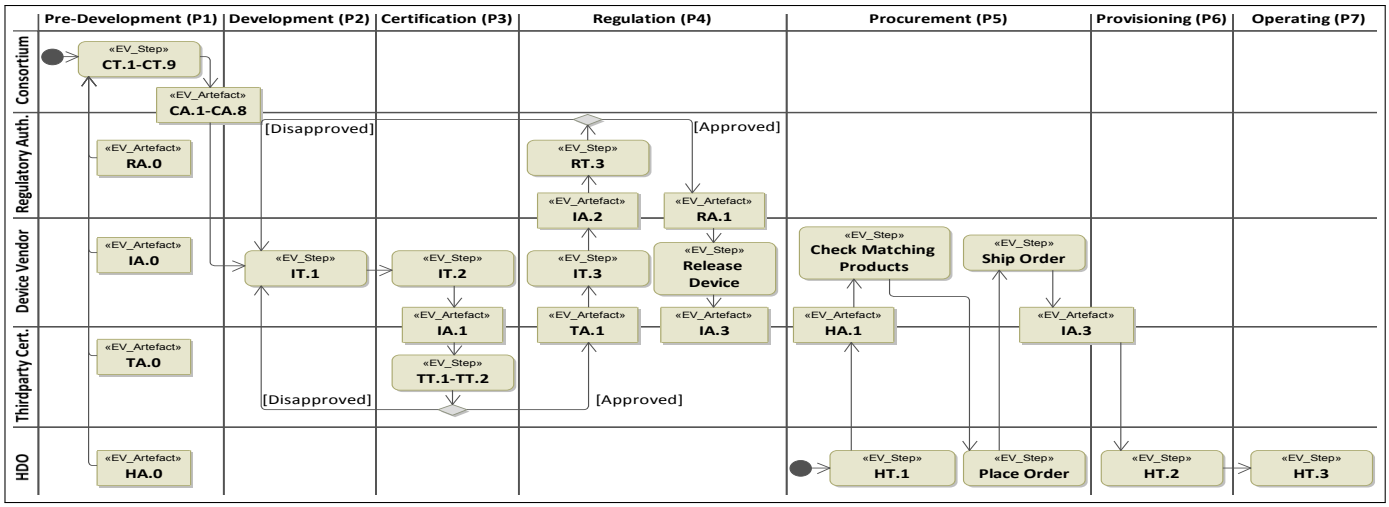

Fig. 2. ICE Device Development Process

by the third-party certification authority. For example, as part of the documentation, vendors use consortium tooling to generate assurance cases based on preliminary compliance testing results. Vendors receive *digital certificates* (TA.1) once this has been completed that confirm the compliance of the submitted components.

IT.3 Submit a *regulatory review submission package* (IA.2). *Regulatory review submission package*s consist of documentation that regulators mandate, the compliance certification (TA.1) from the third-party certification authorities, and possibly the component itself depending on the regulator's policy. After vendors successfully complete third-party testing, they can apply for regulatory review focused on component safety. When regulatory authorities complete their review, they issue *digital certificates* (RA.1) indicating that the components have been reviewed and approved for safety.

Artifacts: In the ICE ecosphere, *digital certificates* ((TA.1) and (RA.1)) are tokens of compliance, safety, and trust of the components. The certificates are acquired by submitting **(IA.1)** *compliance test application package*s to third-party certification authorities and **(IA.2)** *regulatory review submission package*s to regulatory authorities. After completing these processes, vendors offer **(IA.3)** *ICE components* (e.g., *Devices*, *Apps*, and *Platform Components*) equipped with *interface descriptions* of their capabilities and *digital certificates* attesting to their compliance claims.

Dependencies: Vendors must obtain interface and process standards ((CA.1)–(CA.6)), and tooling / supporting resources ((CA.7),(CA.8)) from the consortium. For the final product, vendors must acquire *digital certificates* ((TA.1),(RA.1)) from third-party certification and regulatory authorities.

C. Third-Party Certification Authorities

Description: ICE component vendors are motivated to build products that are compliant to *architecture standards* (CA.1) in order to maximize interoperability with other ICE components. However, they may not have enough resources to perform certain tests. For example, system testing requires that all ICE components are in place; however, individual vendors may not have access to a full suite of components to perform such tests. Third-party certification authorities are organizations that can perform these specialized tests (e.g., system testing) in addition to basic compliance tests; certification authorities are crucial to sustain the integrity of ICE ecosphere because: (1) they are "gate keepers" to ensure that only compatible ICE components are deployed, (2) they provide independent attestation of the claims made by vendors – thus, establishing higher degrees of trust in vendor functional and safety claims, and (3) the compliance testing needed to support regulatory submissions could be performed by third-party certification authorities.

Tasks: Three categories of tests are performed by the third-party certification authority:

TT.1 Perform interface compliance testing in order to verify that the provided component is compliant to its declared interface. A portion of this testing is conducted with the test suite generated by a consortium-approved tool (note that this is identical to an activity performed by vendors). By performing this redundant testing, certification authorities can confirm the claim of vendors that the submitted component is compliant.

TT.2 Perform ICE compliance testing to verify that the component meets relevant ICE standards. This includes safety-related testing based on risk management guidelines (CA.6) and proprietary tests to ensure ICE-compliance of the components. For example, non-functional tests (examining, i.e., security and performance aspects) could be devised, as well as (potentially) destructive testing that injects exceptional and faulty inputs to verify the robustness of the component.

TT.3 Issue *ICE Compliance Certificates* (TA.1). Once a component's claimed compliance is confirmed, the authority issues a *digital certificate* as evidence that the tested component is an effective ICE component [5].

Artifacts: The authority issues **(TA.1)** *ICE Compliance Certificates* with the compliance test report.

Dependencies: Similar to ICE component vendors, the authority depends on interface and process standards ((CA.1)–(CA.6)) in order to perform their tasks. Consortium tooling and supporting resources ((CA.7), (CA.8)) are also required for tests and reviews *compliance test application packages* (IA.1).

D. Regulatory Authorities

Description: Regulatory authorities, in the interest of public health, regulate the safety and effectiveness of medical sys-

tems. Because an ICE system directly impacts the health of humans, these authorities should be involved in regulating MAPs. Due to ICE characteristics (e.g., interchangeable components), ICE will have component-wise regulations; regulators evaluate the safety and effectiveness of each component rather than an instance of an integrated ICE system. Moreover, the authorities recognize standards to foster innovation and grant authority to third-party certification organizations in order to distribute the load of review tasks.

Tasks: The envisioned tasks of the regulatory authority are:

RT.1 Recognize ICE standards and competence of third-party certification authorities for compliance testing and safety assessment of ICE components **(RA.3)**. Regulators may evaluate the capabilities of third-party certification authorities in terms of producing sufficient evidence which will be used by regulators in determining the safety and effectiveness of devices.
RT.2 Create and document *policies* **(RA.2)** for, e.g., classification of ICE components and the digital certificate hierarchy.
RT.3 Review components based on submission artifacts from vendors. These artifacts contain evidence to support claims that the components are safe and effective for their intended use. For example, with consortium-provided tooling, the authority uses the hazard analysis results in the submission documents to identify hazards as inputs for risk assessment. If the review results in approval, a *digital certificate* **(RA.1)** attesting to this fact is issued to the vendor.

Artifacts: Based on ICE components classifications, the authority **(RA.1)** issues a *Safety Review Certificate* as a token of safety review approval, **(RA.2)** provides *policies*, and **(RA.3)** *recognizes the standards and the certifiers*.

Dependencies: The regulatory authority must have the interface / process standards in order to recognize them (**(CA.1)**–**(CA.6)**). Consortium tooling and supporting resources (**(CA.7)**,**(CA.8)**) are used for the reviews of the *regulatory review submission packages* **(IA.2)**.

E. Healthcare Delivery Organizations

Description: Healthcare Delivery Organizations (HDOs) such as hospitals provide healthcare services to patients. HDOs attempt to utilize cost-effective technologies to improve the quality of care and reduce medical errors. In addition to the utility provided by *ICE Apps*, activities in HDOs (e.g., assessment for procurement) could also be supported by ICE device interface and app descriptions.

Tasks: HDOs purchase, configure and operate ICE components to provide healthcare services with ICE systems. HDOs:

HT.1 Assess and procure ICE components based on their needs. With the consortium-approved tooling **(CA.7)**, *specific needs of ICE components* **(HA.1)** could be expressed unambiguously in a machine-readable format and the component vendors could test compatibility with their products, thereby greatly reducing the workload of the (currently manual) process of technology assessment.
HT.2 Provision components according to HDO policies. For example, patient-controlled analgesia (PCA) pump orders may vary among HDOs (e.g., units for drug, kind of drug used), so default settings related to PCA prescriptions (or pump settings) must be set according to HDO practices. Also, access control policies should be enforced by configuring the access privileges of ICE components.
HT.3 Operate deployed ICE systems in various clinical contexts, such as launching and operating necessary *Apps* and *Devices* on the ICE Platform in order to provide *services* to patients. Although the ICE platform itself checks compatibility between *Apps* and *Devices* at launch time, multiple devices may be compatible with an app's requirements; a clinician may select the most suitable device given the clinical situation.

Artifacts: In order to procure ICE components, HDOs require bids for each needed component. The **(HA.1)** *manuscript of needed capabilities* portion of a bid document is described by the ICE standard. Then, HDOs integrate ICE components into their ICE system and provide healthcare services to patients.

Dependencies: HDOs must understand the ICE interface language (**(CA.1)**–**(CA.4)**,**(CA.6)**) in order to specify component capabilities **(HA.1)**, and they require relevant tooling and resources (**(CA.7)**,**(CA.8)**).

IV. Device Product Life Cycle in ICE Ecosphere

In this section we follow an example device (in this case a pulse oximeter) through its lifecycle to illustrate how stakeholders collaborate in the ICE ecosphere. Phases of the life cycle are depicted as vertical swimlanes in Figure 2.

The pulse oximeter non-invasively measures oxygen saturation (SpO_2) level and pulse rate (PR) using a finger clip. It also has a configurable lower-limit (SpO_2 Low Limit) and a time-based alert (SpO_2 Low) for when the SpO_2 value drops lower than the limit for a certain period of time (e.g., 10 sec.).

Predevelopment: In order to facilitate interoperability between *Apps*, *Devices*, and *ICE Platform* components, the consortium develops architecture and interface standards for ICE, associated tooling, and supporting resources (**(CA.1)**–**(CA.8)**) in Phase 1 (P1 in Figure 2). For our example pulse oximeter, the consortium standardizes the nomenclature and data representation for SpO_2 and PR physiological parameters as well as the SpO_2 Low Limit setting and the SpO_2 Low alerts. The consortium also provides *reference models* for pulse oximeter devices that specify basic pulse oximeter capabilities (i.e., reporting SpO_2 and PR), thereby improving the cross-vendor consistency and the reusability of interface models.

Development: In Phase 2 (P2 in Figure 2), the vendor reuses the pulse oximeter *reference model* and extends it by adding the SpO_2 Low Limit setting and SpO_2 Low alert. During development, the vendor also generates test suites from the interface description using the consortium tooling, and conducts compliance testing (e.g., tests of the SpO_2 Low Limit setting, and a measurement report test based on the quality-of-service properties of the reported SpO_2 and PR).

Certification: As the prototype is released, the vendor prepares a *compliance test application package* **(IA.1)** to acquire *ICE compliance certification* **(TA.1)**. This package includes the physical pulse oximeter as well as other documentation (e.g., the compliance test results). In Phase 3 (P3 in Figure 2), the third-party certification authority confirms the vendor's compliance testing results, performs proprietary testing (e.g., robustness testing such as invalid setting for SpO_2 Low Limit

setting), and reviews safety parameters. An *ICE compliance certification* **(TA.1)** is issued when the device passes Phase 3.

Regulation: With the *ICE compliance certification* in-hand, the vendor prepares a *regulatory review submission package* **(IA.2)** in Phase 4 (P4 in Figure 2) for review based on the device's classification. For example, the FDA categorizes most noninvasive pulse oximeters as Class II, consequently requiring submission of premarket notification (510(k)s). The consortium tooling assists in the preparation of submission documentation and highlights safety-related parameters (e.g., the accuracy of reported SpO_2 values, description of visual and audible alarms, etc.) of greatest interest to the authority. The authority then returns the *Safety Review Certificate* **(RA.1)** attesting that the device is safe for intended use. The vendor embeds the acquired digital certificates (**(TA.1)** and **(RA.1)**) into the pulse oximeters and releases the devices to the market.

Procurement: HDOs use the interface modeling language to specify the desired capabilities of a pulse oximeter (instead of, i.e., describing them in natural language) in Phase 5 (P5 in Figure 2). For example, a HDO might compose an interface model describing a pulse oximeter that complies with the reference model SpO_2 Low alert report capability and SpO_2 Low Limit setting capability. The device vendor then runs a compatibility test between the released pulse oximeter interface model and the HDO's interface model in order to verify whether the product meets the needs of the HDO.

Provisioning: If the example pulse oximeters are purchased by a HDO, then that organization must provision the devices based on its policies and workflows in phase 6 (P6 in Figure 2). If, for example, the purchasing HDO is a children's hospital located over 10,000 ft. in altitude, then the SpO_2 Low Limit setting may need to be set lower than 95% in order to reduce nuisance alerts [14].

Operating: After the provisioning of the pulse oximeter is complete, it is deployed and ready for use in Phase 7 (P7 in Figure 2). The pulse oximeter is now ready to be used with, e.g., an *App* that shuts off an infusion of a PCA pump when: (a) the SpO_2 Low alert triggers, (b) the patient's respiratory rate (RR) is low, and (c) her end tidal carbon dioxide ($EtCO_2$) is high. At launch time, a clinician with the necessary access privileges will launch the app. At this time, the *App*'s required device capability description is transferred to the ICE Platform (more specifically the *Network Controller*) in order to identify compatible devices. Part of the *App*'s required capability description specifies that SpO_2 and PR measurements are to be reported at a certain frequency (e.g., every 500ms) and a SpO_2 Low alert is to be produced when the SpO_2 Low Limit is violated. If the example pulse oximeter matches these requirements, the device is listed as a compatible device in the clinician's control panel (e.g., *Supervisor*), and the clinician can launch the *App* when all necessary devices are selected.

Recall: If components are found to be unsafe, hazardous, or otherwise defective after they are released to the market, they are recalled. Products are either fixed or refunded by the seller. Specific criteria and procedures for recall will likely vary according to the regulations of individual countries.

Decommissioning: Components are decommissioned when the HDO replaces the components with other products or irreparable damage occurs. In contrast to a nation-wide recall, the scope of decommissioning is local to the HDO.

V. Conclusion

In this paper, we have described the basic principles of MAP ecospheres and how stakeholders collaborate to sustain them. Some of the concepts in this paper represent new directions that the medical system and standards development communities will need to pursue to develop platform-based medical systems. Many of the principles are motivated by interoperability organizational approaches in other domains such as connectivity (e.g., USB, Wi-Fi, Bluetooth) and mobile platforms (e.g., Android, iOS). Our research is also examining the details of verification techniques for compliance assurance and interface specifications tailored to the MAP vision.

References

[1] FDA Recognized Consensus Standards:ASTM F2761, 2015. Available at http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfstandards/detail.cfm?id=32430.

[2] OMG IDL, 2015. Available at http://www.omg.org/gettingstarted/omg_idl.htm.

[3] ASTM International. ASTM F2761 - Medical Devices and Medical Systems - Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE), 2009.

[4] J. Hatcliff, A. King, I. Lee, A. MacDonald, A. Fernando, M. Robkin, E. Vasserman, S. Weininger, and J. M. Goldman. Rationale and Architecture Principles for Medical Application Platforms. In *International Conference on Cyber-Physical Systems (ICCPS)*, 2012.

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

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

[7] Y. J. Kim. Vision on ICE Ecosphere and Stakeholders. Technical Report SAnToS-TR2015-5, Kansas State University, Department of Computing and Information Sciences, 2015.

[8] Y. J. Kim, J. Hatcliff, V. P. Ranganath, Robby, and S. Weininger. Integrated Clinical Environment Device Model:Stakeholders and High Level Requirements. In *Medical Cyber Physical Systems Workshop (MCPS)*, 2015.

[9] A. King, S. Chen, and I. Lee. The middleware assurance substrate: Enabling strong real-time guarantees in open systems with openflow. In *Object/component/service-oriented realtime distributed computing (ISORC), 17th IEEE Computer Society symposium on*. IEEE, 2014.

[10] 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 *Proceedings of the 31st International Conference on Software Engineering*, 2009.

[11] A. L. King, L. Feng, S. Procter, S. Chen, O. Sokolsky, J. Hatcliff, and I. Lee. Towards Assurance for Plug & Play Medical Systems. In *Computer Safety, Reliability, and Security*. Springer, 2015.

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

[13] S. Procter, J. Hatcliff, and Robby. Towards an AADL-Based Definition of App Architectures for Medical Application Platforms. In *Proceedings of the International Workshop on Software Engineering in Healthcare*, Washington, DC, July 2014.

[14] S. Shrestha, S. Shrestha, L. Shrestha, and N. Bhandary. Oxygen saturation of hemoglobin in healthy children of 2-14 years at high altitude in Nepal. *Kathmandu Univ Med J (KUMJ)*, Jan-Mar 2012.