OR.NET: A Service-Oriented Architecture for Safe and Dynamic Medical Device Interoperability

Martin Kasparick[1], Malte Schmitz[2], Björn Andersen[3], Max Rockstroh[4], Stefan Franke[4], Stefan Schlichting[5], Frank Golatowski[1], and Dirk Timmermann[1]

***Abstract*— Modern surgical departments are characterized by a high degree of automation supporting complex procedures. It recently became apparent that integrated operating rooms can improve the quality of care, simplify clinical workflows, and mitigate equipment-related incidents and human errors. Particularly using computer assistance based on data from integrated surgical devices is a promising opportunity. However, the lack of manufacturer-independent interoperability often prevents the deployment of collaborative assistive systems. The German flagship project OR.NET has therefore developed, implemented, validated, and standardized concepts for open medical device interoperability. This paper describes the universal OR.NET interoperability concept enabling a safe and dynamic manufacturer-independent interconnection of point-of-care medical devices in the operating room and the whole clinic. It is based on a protocol specifically addressing the requirements of device-to-device communication, yet also provides solutions for connecting the clinical IT infrastructure. We present the concept of a Service-Oriented Medical Device Architecture (SOMDA) as well as an introduction to the technical specification implementing the SOMDA paradigm, currently being standardized within the IEEE 11073 SDC series. In addition, the *Session* concept is introduced as a key enabler for safe device interconnection in highly dynamic ensembles of networked medical devices; and finally, the security of a SOMDA is discussed.**

I. Introduction

Medical treatments, diagnosis, and care procedures as well as the used medical devices have become more and more complex. One of the most critical environments is the operating room (OR) [1], [2]. Equipment-related incidents and surgical errors have a significant proportion [3], [4], [5]. Integrated ORs have been identified as a promising concept to meet the complexity induced challenges [6], [7]. But "as we know, interoperability is an almost non-existent feature of medical devices" [8], especially not for multi-manufacturer systems. While this has been stated by Lesh et al. in 2007, the statement is still appropriate. The OR.NET [9] project on safe and dynamic networking in OR and clinic aimed to overcome this lack of interoperability.

Various research projects have been working in the field of medical device interoperability and OR integration based on open standards like the US-American MD PnP project [10] or the SCOT project in Japan [11]. In Germany the flagship project OR.NET funded by the Federal Ministry of Education and Research (BMBF) has been working on a comprehensive interconnection of medical devices among each other within OR and clinic as well as to the clinical information systems, based on a Service-Oriented Architecture (SOA). While the pre-projects smartOR [12] and DOOP [13] mainly focused on the technical realization, OR.NET has been working on a wide spread of issues: The medical device interoperability has been improved, regulatory issues have been discussed and strategies for risk management and approval have been developed being suitable for systems of dynamically interconnected medical devices [14], [15], new concepts for usable human-machine-interaction have been developed [16], [17], [18], operator strategies have been focused [19], standardization process has been started, and concept validation by comprehensive demonstrators has been conducted [16], [20].

Some of the leading manufacturers have solutions for integrated ORs on the market (see Section II-B). Such solutions lack of flexibility as only devices of the specific manufacturer – respectively a small subset of available devices – can be integrated. Thus, it is not possible to use the technically best device for the specific use case and to buy the device with the best price-performance ratio in many cases.

An integrated OR based in multi-manufacturer interoperability will improve workflow during the surgical treatment as well as beyond the OR to increase patient's safety and save money. For example, intelligent alert systems will reduce false alarms resulting in a higher probability of recognizing and handling serious alarms; partial automated documentation will reduce the workload of the clinical staff; intelligent planning systems for surgical treatments will increase degree of capacity utilization and efficiency of resources like the ORs and medical devices; etc.

Within the OR.NET project, an enabling technology has been developed for manufacturer-independent interoperability and the open integrated OR of the future. While the presented use cases focus on the OR, the developed concepts and implementations are not limited to the OR and can be used for all Point-of-Care (PoC) medical devices. This paper presents an overview of the OR.NET interconnection

[1]Author is with the Institute of Applied Microelectronics and Computer Engineering, University of Rostock, 18119 Rostock, Germany firstname.lastname@uni-rostock.de

[2]Author is with the Institute for Software Engineering and Programming Languages, University of Lübeck, 23562 Lübeck, Germany firstname.lastname@isp.uni-luebeck.de

[3]Author is with the Institute of Medical Informatics, University of Lübeck, D-23562 Lübeck, Germany andersen@imi.uni-luebeck.de

[4]Author is with the Innovation Center Computer Assisted Surgery (ICCAS), University of Leipzig, 04103 Leipzig, Germany firstname.lastname@medizin.uni-leipzig.de

[5]Author is with Drägerwerk AG & Co. KGaA, 23558 Lübeck, Germany firstname.lastname@draeger.com

architecture (Section III). As the focus of this paper is on the device-to-device communication and interoperability, Section IV introduces the concept of a Service-Oriented Medical Device Architecture (SOMDA). Section V presents a specification realizing the SOMDA paradigm that is currently in the process of standardization, called IEEE 11073 SDC (Service-oriented Device Connectivity) family. Within the OR.NET project, SDC has also been called Open Surgical Communication Protocol (OSCP). Safety and security issues are discussed in Section VI and Section VII. In Section VIII, a brief overview of developed demonstrators for concept validation is presented.

II. State of the Art

In this section, we give a brief overview of the state of the art concerning integrated OR solutions and research projects on medical device interoperability, after presenting a definition of the term *interoperability*.

A. Interoperability Definition

The HIMSS dictionary defines interoperability of health information systems as the ability "to work together within and across organizational boundaries in order to advance the effective delivery of healthcare for individuals and communities" [21]. Three types of interoperability have been defined [22]:

- Foundational interoperability: ability to exchange data
- Structural (or syntactical) interoperability: structure or format of the exchanged data
- Semantic interoperability: ability of two or more systems to exchange information, interpret this information correctly, and use this information

B. Available Integrated OR Solutions

Solutions for integrated ORs are available on the market from some of the leading medical manufacturers. According to a report of iData Research [23] the leading competitors for integrated ORs in the US market are Stryker (iSuite™), Karl Storz (OR1™), STERIS (Harmony iQ®), and others.

The available solutions are monolithic systems using proprietary protocols and hardware for data exchange. Open standards for a manufacturer-independent interoperability of medical devices within the OR are not in the scope. This leads to less flexibility for clinic operators and physicians. After the decision for one manufacturer, there is a high dependency on the available devices for the chosen system. Integrating other devices offering for example better technical solutions for the specific use case, new innovations, or better price-performance ratio is not possible in the most cases. Thus, potentially not the best equipment is used for the specific surgical treatments and high costs are incurred for the operators.

C. Research Projects on Medical Device Interoperability

Interoperability of medical devices among each other and with the clinical information systems is an important research topic. The vision of an integrated multi-manufacturer OR based open standards is emerging to overcome proprietary and isolated solutions of single-manufacturers.

Since 2004 the "Medical Device 'Plug-and-Play' Interoperability Program (MD PnP)" [10] has been working on the definition, development, and implementation of an "Integrated Clinical Environment (ICE)". The multi-institutional community is led by the Massachusetts General Hospital (MGH), USA. The "General requirements and conceptual model" of an ICE was standardized as the first part of the ASTM standard F2761-09(2013) [24] titled "Medical Devices and Medical System – Essential principles of safety and performance for equipment comprising the patient-centric integrated clinical environment (ICE)". Further parts of this standard are planned. The ICE describes a concept for an interconnected OR. The concrete implementation is not defined. Note, the IEEE 11073 SDC specification (see Section V) holds the requirements defined in the ICE standard. The MD PnP consortium uses the Data Distribution Service (DDS) [25] standard to implement the medical device interconnection. DDS is an implementation of the Service-Oriented Architecture (SOA) paradigm using the publisher-subscriber communication pattern. An open-source framework is provided, called OpenICE [26].

In Japan the research project "Smart Cyber Operating Theater (SCOT)" works on the development of an integrated OR. A standard from the factory automation domain called "Open Robot/Resource interface for the Network (ORiN)" is used to build up the so-called OPeLiNK middleware system [11].

Several German research projects worked in the field of multi-manufacturer medical device interoperability. Previous projects of OR.NET are for example the project smartOR [12] (based on the projects FUSION and orthoMIT [27]), the TiCoLi communication library [28] or the projects DOOP [13] (based on the project TeKoMed [29]). The SOA paradigm has been figured out as a suitable basic concept for an interoperable medical device interconnection. While the previous German projects have shown the technical and multi-vendor feasibility, OR.NET made technical enhancements and refinements, focused on standardization, regulatory, and approval issues, and did a validation of the concepts.

III. The overall OR.NET Communication Approach

The OR.NET project aimed for a comprehensive and interoperable interconnection of the medical devices within the OR among each other as well as between medical information systems and medical devices. A schematic overview is given in Fig. 1. In the following subsections, we will explain the concepts and ideas in more detail.

A. Medical Device-to-Device Communication

As a suitable basic concept for the medical device-to-device communication, the service-oriented architecture (SOA) was carved out. Since the OR.NET project, the

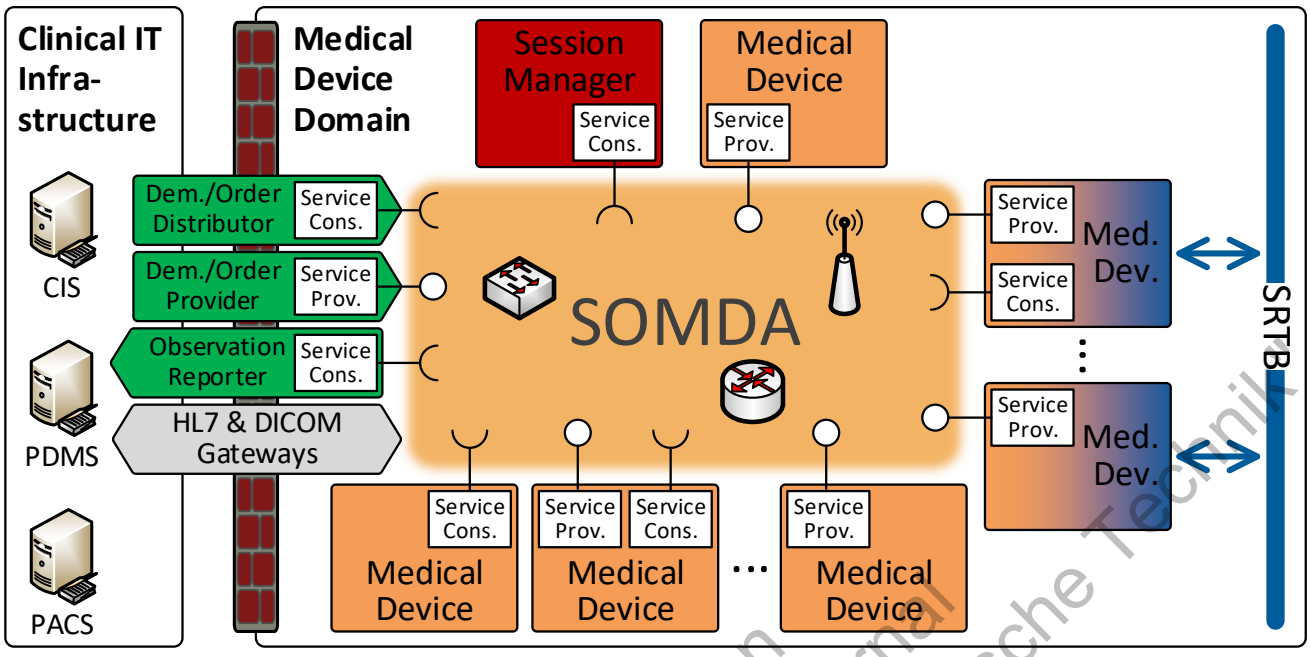

Fig. 1. The overall OR.NET communication approach, incorporating *SOMDA*-based medical device interoperability (orange), *SRTB* real-time communication (blue), and interconnection to the clinical IT infrastructure (green and gray). *Medical Devices* (orange) can implement service provider and/or service consumer functionalities. *Medical Devices* can be connected to both, *SOMDA*- and *SRTB*-network (orange and blue). The *Session Manager* (red) is a dedicated participant (see Section VI). Abbreviations: SOMDA – Service Oriented Medical Device Architecture; SRTB – Surgical Real-Time Bus; Med. Dev. – Medical Device; Service Prov. – Service Provider; Service Cons. – Service Consumer; Dem. – Demographics.

SOA has been refined to a Service-Oriented Medical Device Architecture (SOMDA) (see Section IV). A technical specification implementing the SOMDA paradigm has been developed and is currently in the process of standardization as IEEE 11073 SDC family, or short SDC (for Service-oriented Device Connectivity, see Section V). Note, an equivalent name for SDC, used during the OR.NET project, is Open Surgical Communication Protocol (OSCP). The communication is based on standard network hardware and protocols like Ethernet (IEEE 802.3) or WiFi (IEEE 802.11). This reduces installation and maintenance costs and supports interoperability. As standard Ethernet is not hard real-time capable the medical device network has to be divided into two parts: 1.) The service-oriented interconnection (SOMDA), realized using the technical specification IEEE 11073 SDC based on standard network technology. 2.) The hard real-time capable network, called Surgical Real-Time Bus (SRTB) [30]. It is based in real-time Ethernet systems. Hard real-time capabilities with low latency (e.g., < 10 ms) are necessary for some medical use cases. For example, closed loop scenarios, like the interconnection between a navigation system and a surgical drill in case of a navigated surgery have hard real-time requirements. Investigations during the OR.NET project have shown that the majority of exchanged data (e.g., vital signs, device parameters, etc.) and remote control commands have no or only soft real-time requirements. Thus, the majority of use cases can be realized using the highly flexible and cost efficient SOMDA network.

The middle and right parts of Fig. 1 visualize the medical device-to-device communication. The middle part illustrates the SOMDA network. The right most medical devices are connected both to the SOMDA network and the real-time capable SRTB network. The concept of connecting medical devices to both networks allows the combination of the advantages of both worlds: flexible, plug-and-play capable, cost efficient interconnection for most kinds of vital signs and parameter exchange and remote configuration of device parameters and highly reliable, deterministic, low-latency interconnection for safety critical commands. Although the medical devices are part of both networks, the networks are separated from each other in a physical or logical (e.g., VLAN) manner.

B. Medical Device-to-Infrastructure Communication

In addition to the two newly-introduced networks for soft and hard real-time medical device-to-device communication, the clinical IT infrastructure needs to be considered. Due to safety and security considerations, there must be a strict separation between these networks (see Sect. VII). The left domain in Fig. 1 contains the typical clinical IT infrastructure components such as Clinical Information System (CIS), Patient Data Management System (PDMS), or Picture Archiving and Communication System (PACS).

In spite of this separation, information has to be exchanged between both domains. Examples include patient demographic data, order information, preoperative diagnostics, and laboratory findings that shall be displayed within the OR to support the clinical staff or may be used by the medical devices to load specific configurations. When the OR

devices become aware of the user-confirmed association with a patient, sending documentary data back to the information systems can be partially automated and thereby significantly reduce the workload of the physicians. Another common use case is the exchange of image data. On the one hand, preoperative data can be used during surgery. On the other, images and video sequences from an endoscope or microscope that are recorded during a procedure shall be stored in the PACS for documentary and/or teaching purposes.

Within the OR.NET project, concepts have been developed and components implemented to overcome this gap without violating the strict network separation constraint.

The IEEE 11073 SDC aims for competing with neither the established communication standards for information systems such as Health Level Seven (HL7) and Digital Imaging and Communications in Medicine (DICOM) nor with other emerging standards such as the HL7 Fast Healthcare Interoperability Resources (FHIR) [31], but rather to complement them. Gateways for HL7 and DICOM have thus been developed to connect medical devices that have sufficient resources to implement either of these communication stacks in addition to SDC. These gateways operate as dedicated network transitions, depicted as a gray box in the left part of Fig. 1. With regard to safety and security, a small number of network transitions is much more effectively and efficiently manageable than a multitude of direct connections between information systems and medical devices.

Many medical devices within the SOMDA are resource-constrained so that DICOM or HL7 communication can either not be implemented or the cost-benefit ratio for the implementation is too unprofitable for the manufacturer. Nevertheless, basic patient demographic and order information is highly useful for many devices and applications in order to offer functionality such as automated documentation or the patient-, physician-, and procedure-specific automated parameter configuration of a device. Therefore, components have been developed and implemented within the OR.NET project that translate patient demographic data and order information from an HL7 representation into the IEEE 11073 SDC representation, thus being available in the SOMDA. This has the advantage that only one communication stack has to be implemented on the device to interact with other medical devices and to receive information from the IT systems. In the left part of Fig. 1 these components are illustrated as green boxes bridging the domain gap.

The translation and propagation of data from the clinical IT systems to the SOMDA participants is specified as a connector that can be implemented in two distinct ways: A *Demographics and Order Distributor* pushes the information to all devices that provide a corresponding remote control operation for the *Patient Context* (containing basic patient demographics) and *Workflow Context* (containing basic order information). Only devices that have use for receiving this data will need to implement these remote control operations. In contrast, with the implementation of a *Demographics and Order Provider*, interested devices can invoke the corresponding retrieval services of the data provider to get acquire the translated information.

For the reverse direction, an *Observation Reporter* is specified that transforms and forwards physiological data, alerts, device parameters, etc. from the medical devices to the clinical information systems. Advanced implementations of a reporter can aggregate and filter this data or contain analytic capability to derive clinical findings from the raw input.

C. Scope of the Paper

This paper concentrates on the service-oriented medical device-to-device communication, while real-time aspects are described in Pfeiffer et al. [30] and [32]. The device-to-infrastructure communication (detailed information can be found at Andersen et al. [33]) and ongoing topics like approval (see Janß et al. [34]), usability, operator strategies, etc. are not in the scope of this paper.

IV. From SOA and SODA to SOMDA

In this Section, we describe the evolution of refining the paradigm Service-Oriented Medical Device Architecture (SOMDA) from the Service-Oriented Device Architecture (SODA) based on the Service-Oriented Architecture (SOA).

A. Service-Oriented Architecture (SOA)

The Service-Oriented Architecture (SOA) is a design principle representing heterogeneous and distributed capabilities, like methods or applications, as services that are platform- and application-independent. Currently, there is no consistent definition of a SOA. Thus, we will refer to the widely accepted basic ideas based on [35], [36], [37], [38]. The basic characteristics of a SOA are the *loose coupling* requiring *discoverability* of services, as well as standardized *service descriptions* and *service contracts*, *service abstraction*, *service reusability*, *service autonomy*, *service composability*, and *statelessness*.

There are three different types of participants of a SOA. (See also upper part of Fig. 2.)

- service provider
- service consumer
- service registry/broker

Service providers offer their capabilities via services. The service consumers can use these provided services to meet their needs. A service consumer, or also called client, does not provide any services or information. Note, it is possible that both roles, service provider and service consumer, are implemented on one physical instance. Logically, there is a strict separation between both roles.

The service registry or broker stores references of provided services. Therefore the service providers publish their services to the service registry. A service consumer asks the service registry for suitable services and get the corresponding references if available. Afterwards, a service consumer binds dynamically to the services and interacts with them.

The most common realization of the SOA paradigm are Web Services. The W3C Web Service Glossary [39] defines

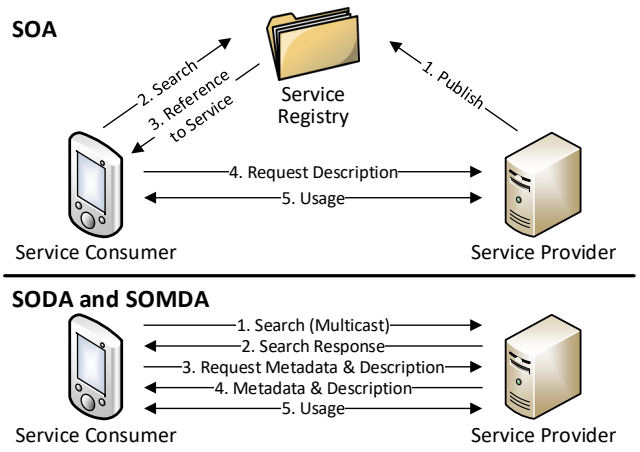

Fig. 2. Discovery process illustration for SOA and SODA/SOMDA (upper part derived from [38]).

Web Services as software systems "designed to support interoperable machine-to-machine interaction over a network." Web Services describe their interfaces using the Web Service Definition Language (WSDL) [40], [41]. Typically, Web Services use a UDDI registry (Universal Description Discovery & Integration) [42]. The interaction between Web Services and the service consumers is based on SOAP messages [43].

B. Service-Oriented Device Architecture (SODA)

The advantages of flexibility, plug-and-play capabilities, standardized and self-descriptive interfaces, low implementation and maintenance effort, etc. of a SOA became also interesting for the interconnection between (resource-constrained) physical devices, like sensors and actuators. De Deugd et al. [44] introduced the concept of a Service-Oriented Device Architecture (SODA). The basic idea is to model devices, respectively their capabilities, as services. The SODA aims on the interconnection of the devices among each other (horizontal integration) as well as on integrating a wide range of physical devices into distributed IT enterprise systems (vertical integration). Mauro et al. [45] introduce and refine patterns for the SODA in general (*Service Encapsulation*, *Legacy Wrapper*, *Dynamical Adapter*, and *Auto-Publishing*) and discuss them for the healthcare domain.

Realizing the SOA paradigm for the device interconnection leads to the advantage of lower implementation and maintenance effort. This is due to the explained characteristics, like loose coupling, reusability, or composability. All components and functionalities can be implemented independently and as simple as possible. Thus, the SODA is supposed to reduce the system's complexity.

The Devices Profile for Web Services (DPWS) [46] is one specific realization of the SODA concept. DPWS is a suitable implementation of Web Services for resource-constrained devices. Therefore, DPWS uses several Web Service standards, so-called WS-* standards, like WS-Discovery, WS-Eventing, WS-Security, etc. To meet the requirements of embedded devices subsets, restrictions, concretions, and extensions of these WS-* standards are defined. As DPWS uses the Web Services Dynamic Discovery (WS-Discovery) [47] specification there is no need for a centralized service registry (e.g., UDDI). The discovery process can be realized in an explicit or implicit way. Explicit discovery means that the service consumers search for services using UDP-Multicast *Probe* messages. Every suitable service will answer with a *ProbeMatch* message. (An abstract visualization is shown in Fig. 2.) If a service provider joins the SODA environment it sends an UDP-Multicast *Hello* message and analogous a *Bye* message if it leaves. This is called implicit discovery. Note, DPWS does not prohibit a centralized discovery proxy.

C. Service-Oriented Medical Device Architecture (SOMDA)

Within the OR.NET project, the concept of the SODA has been refined to a *Service-Oriented Medical Device Architecture (SOMDA)*, which requires more than just a SODA for medical devices. The SOMDA thus exhibits some specific characteristics that take into account the safety requirements of systems of networked medical devices:

(1) standardized data description extending and complementing the interface description for medical device interoperability,
(2) patient safety considerations,
(3) the capability of forming subsets/ensembles of network participants, and
(4) regulatory issues pertinent to medical devices.

A standardized interface description, which is one of the basic principles of SOA and SODA, is not sufficient for dynamic medical device interconnection. In addition, a standardized way to describe the provided and exchanged data is necessary to ensure interoperability through safe and correct data interpretation. Beyond the description of a device's own capabilities and device state, this also includes, for example, the possibility of modelling measurement quality or intended use of a value.

Data exchange, especially remote control, is highly safety-critical in medical device systems. Compared to other SODA applications, the safety requirements are thus higher in a SOMDA. The most common example is the demand for single-fault safety in many medical remote control use cases. Dual channel transmission is the state-of-the-art solution in the medical device domain, which a SOMDA consequently has to provide. Furthermore, the same information can have various meanings in different contexts. Thus, a SOMDA-compliant implementation shall also provide a mechanism for transmitting safety-related contextual information to ensure a safe interpretation of the transferred data within the given context.

The third aspect addresses the demand for high flexibility of dynamic medical device ensembles. In many industrial application scenarios of a SODA, the type and number of devices interacting with each other are known at the time of deployment and typically do not change at runtime. In the clinical environment, however, the device ensembles differ between various kinds of interventions or even between different patients. Additionally, devices are physically moved between ORs and/or used to treat several patients in a

TABLE I

OVERVIEW IEEE 11073 FAMILY OF STANDARDS (SELECTION).

IEEE Standard	Content
Existing	
11073-10101	Nomenclature
11073-10201	Domain Information Model (DIM)
11073-20101	Application Profile, Communication Model
11073-30XXX	Transport Profiles
New SDC Family	
P11073-10207	Domain Information & Service Model
P11073-20701	Architecture & Binding
11073-20702	Medical DPWS
To Be Extended	
11073-1010X	Nomenclatures
11073-103XX	Point-of-Care Device Specialization
11073-104XX	Personal Health Device Specialization

short period of time. Consequently, groupings are necessary to organize complex communication structures. A SOMDA thus needs to have the capability to form flexible subsets or *ensembles* of participants. SOMDA ensembles can have arbitrary complexity, from functional units of only two devices up to complex device systems for an intricate surgical procedure. They can be realized hierarchically, meaning that different functional units or other types of device ensembles may form another higher level ensemble. SOMDA participants can join and leave these dynamically. How ensemble management can be implemented using SDC to ensure safe communication between members without interference issues is described in Section VI.

Whereas the first three SOMDA characteristics focus on technical aspects, the fourth takes regulatory considerations into account, as the market of medical devices is highly regulated. This issue cannot be sharply separated from, yet highly influences the previous three aspects. For example, every medical device has to be identifiable using a worldwide unique identifier its vendor assigned to it. The existing Unique Device Identification (UDI) can be used for that purpose. With this unique device identifier, devices can be re-identified in the discovery process even after re-assignment of low-level network addresses. As this example illustrates, regulatory requirements have to be addressed as an overarching issue on all levels of a SOMDA implementation.

V. SOMDA REALIZATION AND STANDARDIZATION: THE IEEE 11073 SDC FAMILY

The technical realization of the SOMDA has to ensure foundational, structural, and semantic interoperability of the medical devices, as described in Section II-A. Therefore, three new standards have been developed addressing the first two interoperability levels and enable the third level:

- IEEE 11073-20702 (Medical DPWS)
- IEEE P11073-10207 (Domain-Information- and Service-Model)
- IEEE P11073-20701 (Service-Oriented Medical Device Exchange Architecture and Protocol Binding)

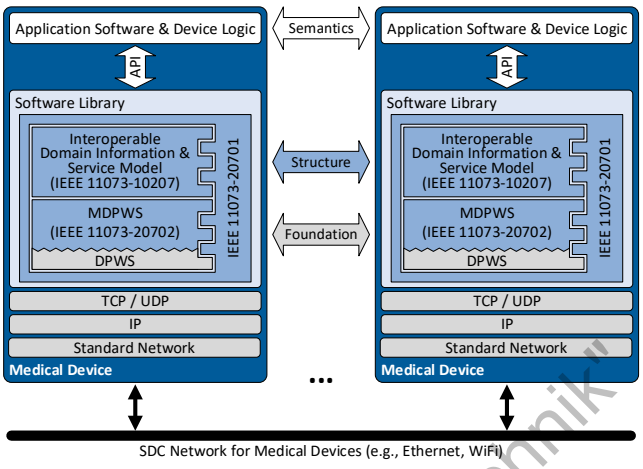

Fig. 3. Visualization of the interconnection between the IEEE 11073 SDC standards and embedment into an example software stack. (Figure derived from [50], [51].)

The interconnection between these standards is displayed in Fig. 3. The Medical DPWS (IEEE 11073-20702) as transport mechanism is independent from the data that is exchanged (IEEE P11073-10207). The allover architecture and the binding between the two formerly mentioned standards are defined in IEEE P11073-20701. Currently, the content of the standards IEEE 11073-10207 and -20701 is in the process of standardization to be added as new parts of the well-known IEEE 11073 family of standards. Thus, the currently correct labeling is IEEE P11073-10207 and IEEE P11073-20701, respectively. When standardization process has been finished, the "P" indicating the proposal state will be omitted. A (selective) overview of the IEEE 11073 standards is given in Table I. These three standards together are called IEEE 11073 SDC family, or short SDC (for Service-oriented Device Connectivity). Note, while SDC is the international term used in the standardization process, an equivalent name for the protocol family is Open Surgical Communication Protocol (OSCP), mainly used during the OR.NET project.

SDC describes the specification for a completely distributed middleware. It enables manufacturer-independent medical device interoperability without any centralized communication components, like managers for discovery, eventing, or service registry. In this section we present a brief introduction into the SDC specification. For further information see [48], [49], [50].

A. Medical DPWS

The Medical Devices Communication Profile for Web Services (MDPWS) realizes the data exchange between the medical devices. It ensures the fundamental interoperability. For the development of MDPWS an existing OASIS standard was used: The Devices Profile for Web Services (DPWS) [46], that is an implementation of the SOA paradigm designed for resource-constrained devices. DPWS provides the ability to exchange data between multiple systems according to the request-response and publish-subscribe patterns. Additionally, dynamic discovery mechanisms are available. There-

fore, DPWS uses the OASIS standard WS-Discovery [47]. It allows explicit discovery (service consumer searches for service providers and services) and implicit discovery (service providers announce itself). Thus, DPWS is basically suitable to realize the communication for an interoperable system of medical devices. Additionally, using an application layer protocol has the advantage that the communication is independent from the underlying network. Typically, standard Ethernet will be used, but also wireless networks are possible. In general, the requirement is that UDP and TCP are available to transmit SOAP [43] messages.

For the specific medical requirements and safety issues MDPWS defines extensions and also some restrictions of DPWS. The main issues are:

- Safe data transmission
- Data streaming
- Compact data transmission

To ensure the interoperability the MDPWS extension specifications have two aspects: The advertisement of the properties, capabilities, and requirements of the medical device and the definition how the information is transmitted. The advertisement is necessary because not every medical device will make use of every MDPWS capability. For example, data streaming will typically be used by devices that produce waveforms, but most other devices do not need this functionality. The basic idea is to include the advertisement into the WSDL (Web Services Description Language) [41] of the web services using WS-Policy [52] mechanisms. The WSDL contains the self-description of the web services provided by the medical device. Including information about additional properties, capabilities, or requirements into the WSDL enables the exchange of this information at runtime. This ensures the foundational interoperability, although devices with different properties operate together in an SDC network.

Safe data transmission is essential for networked medical device systems. Especially, remote control commands have to be transmitted safely. Therefore, MDPWS defines a dual channel transmission and the so-called safety context. The first enables single fault safe data transmission using one physical transmission medium. (For further, information and implementation recommendations see [48].) The second (safety context) allows the transmission of additional safety relevant contextual information within the header of remote control commands. If the safety context does not contain the correct corresponding information, the service provider can reject the command from the service consumer. For example, an OR-table could require that a service consumer that likes to change the height of the OR-table has to include the unit of the value by which the height should be adjusted. This ensures that there is no confusion between cm, mm, in, thou, etc.

The data streaming mechanism is used to transmit waveforms, like ECG or EEG. Therefore, an UDP-based data transmission is defined. Note, DPWS typically uses TCP-based transmissions for data exchange, apart from some parts of the discovery process. The transmission of audio or video streams via MDPWS is not intended.

DPWS uses SOAP to transmit the information. As SOAP uses an XML representation of the information, the amount of data can become (unnecessarily) high. Thus, MDPWS allows the usage of the Efficient XML Interchange (EXI) format [53]. EXI can reduce the message size significantly. This reduces the network traffic and enables lower latency.

As the first of the three standards, MDPWS has passed the standardization process. It is available as IEEE 11073-20702-2016 [54]. A more detailed description of the technical backgrounds of MDPWS can be found in [50].

B. Domain Information & Service Model

The proposed standard IEEE P11073-10207 "Domain Information & Service Model for Service-Oriented Point-of-Care Medical Device Communication" [55] consists of two parts: The Domain Information Model (DIM) describes the structure of the exchanged data. The Service Model defines how to get access to the data. Thus, this standard addresses the structural interoperability.

1) Domain Information Model (DIM): The new DIM is derived from the classical IEEE 11073-10201 DIM [56]. Extensions and changes had to be made to meet the requirements of systems of networked medical devices with multiple connections among each other, based on a service-oriented approach.

The DIM defines a separation between medical device description (*MdDescription*) and state (*MdState*). Both are stored in the so-called *Medical Device Information Base (MDIB)*. The device description contains the self-description of the properties of the medical device with all capabilities that are offered to other SDC participants. (See left part of Fig. 4.) The state describes the concrete values at a certain point of time. (See right part of Fig. 4.) Compared to the device state, the device description is relatively static during runtime, but it can be changed if necessary. Typically, other SDC service consumers will read the device description, respectively the relevant parts, of an SDC device while establishing the connection to the device. During the interaction phase, the SDC consumer will be interested mainly in the device state. The device state containing the concrete values can change very dynamically. For example, the rotation speed of a surgical motor system or the vital signs of a patient monitor change very frequently.

The device description is organized as a tree hierarchy with a height of four, as displayed in the left part Fig. 4. Measurements, settings, calculations, status, etc. of a medical device are called metrics. Metrics are the leaves of the description tree. The type of the metric is described by coded values that belong to coding systems. As default coding system the IEEE 11073-10101 nomenclature [57] is used. For example, a pulse oximeter would have a metric for each pulse and oxygen saturation. The code for the pulse is 149530 (2::18458 or MDC_PULS_OXIM_PULS_RATE) and for the oxygen saturation 150456 (2::19384 or MDC_PULS_OXIM_SAT_O2). Additionally, every metric description contains the unit.

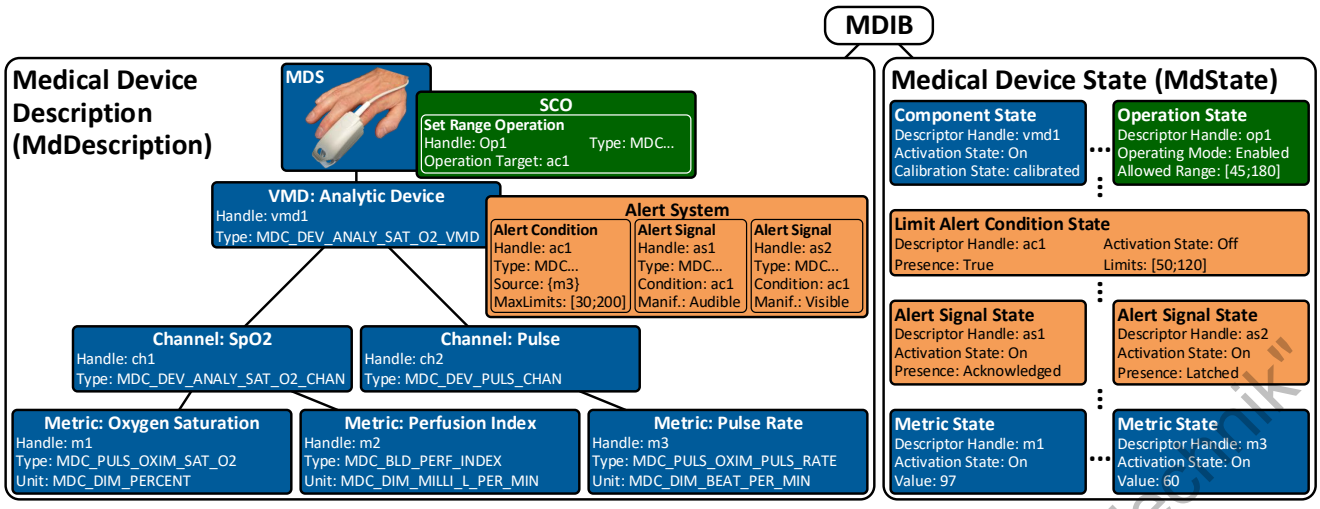

Fig. 4. Example MDIB for a pulse oximeter with reduced complexity of the *MdDescription*. For the *MdState* only a representative sub-set of all corresponding states is illustrated. Colors: Blue – device description tree and corresponding states; Green – SCO and corresponding state; Orange – alert system and corresponding states. Abbreviations: MDIB – Medical Device Information Base; MDS – Medical Device System; VMD – Virtual Medical Device; SCO – Service and Control Object; Manif. – Manifestation.

The unit is also specified as a coded value. For the given example the corresponding unit codes are 264864 (4::2720 or MDC_DIM_BEAT_PER_MIN) or 262688 (4::544 or MDC_DIM_PERCENT), respectively. Furthermore, information like the intended use, the availability, measurement/calculation delays, body side, etc. can be included in the metric description.

The next higher layer of the description tree is formed by the Channels. Channels define groupings (logical or physical) of metrics. Multiple Channels can belong to a Virtual Medical Device (VMD). VMDs are subsystems of Medical Device Systems (MDS). The MDS builds the root node of the tree. For the very simple example of a pulse oximeter, the MDS would only contain one VMD and maybe two different Channels, where each contains one of the pulse and oxygen saturation metrics. For more complex devices, like a patient monitor, the tree will be wider and more complex. For example, the MDS would contain different VMDs for pulse oximeter, ECG, blood pressure, etc.

Alert systems can be defined on the level of MDS, VMD and Channel. An alert system consists of alert conditions and alert signals. The alert conditions monitor physiological or technical aspects. Several alert signals can belong to an alert condition. This allows that an alert can be signaled in different ways: visible, audible, or tangible.

For an SDC device, different contexts can be defined. In contrast to metrics describing device capabilities, *contexts* describe the relationship to the device's environment. Hence, contexts are not part of the description tree but an optional element attached to the MDS node. Furthermore, contexts are not as generic as metrics, because they do not differ much between devices and can be defined more precisely. The location context, for example, can be important for moving devices and the ensemble context can be used to define groupings of devices (e.g., the so-called *session context*, see Section VI); the patient context may contain basic patient demographic information whereas the workflow context can contain information about the encounter and order as commonly issued by clinical information systems. The latter two context descriptors are specifically designed to interconnect the SDC devices to the clinical IT infrastructure. Some contexts can occur several times, e.g., one device can belong to multiple ensembles, like to the device ensemble of the manufacturer, to the general surgical device ensemble, to the endoscopic device ensemble, the endoscopic camera and light source could form a sub-ensemble, etc. The contexts can have different states of their association, like not associated, pre-associated, associated, and disassociated. This allows a fine-grained representation of the contexts over time.

The general concept of contextual information thus allows for an intuitive semantic understanding of an MDS as a *device*, all of whose components (VMDs) *always* operate within the *same* context.

The mechanisms introduced above describe the information an SDC device provides for reading access. For the description of remote control functionality, the Service and Control Object (SCO) can be added to the MDS. Within the SCO several operations can be defined. Basically, these are set operations and activate operations. Set operations are used to manipulate device parameters like settings or alert limits. An active operation can trigger a function of arbitrary complexity at the device, like a simple increase or decrease of parameters or complex reconfigurations of the device system. For further information: Mechanisms to ensure a safe remote activation of device functionalities using an potentially unreliable network are detailed described in [58]. IEEE 11073 SDC is also a key enabler for highly flexible association of control devices and remotely controlled devices [59].

To be identifiable, every element of the description tree has a (device-wide) unique handle. To describe the semantics,

every element of the description tree is tagged with a coded value, as described above for the metrics. This includes the alerts as well as the operations. As the quite simple nomenclature of the IEEE 11073-1010x series is used as the default coding system, more complex and more meaningful systems like LOINC, SNOMET CT, etc. are also possible. Based on the structural interoperability, the comprehensive usage of coded values to specify every element of the SDC device description enables the semantic interoperability.

The second part of the MDIB is the device state. (See right part of Fig. 4.) The *MdState* is not organized as a tree like the *MdDescription*. It is a set of state elements. An explicit hierarchy is not necessary, because it is given by the device description. For every element of the *MdDescription* there is (at least) one corresponding state element in the *MdState*. The reference between state and description is done by the (device-wide) unique description handles. The states contain the concrete values. Descriptive information like the semantics or the unit of a value is not necessary as it is implicitly given by the description. This strict separation reduces computational effort and network traffic, because during runtime the description will typically be relatively static and the state will change frequently, as already discussed. Thus, keeping the state information as simple as possible reduces the effort for the devices.

Depending on the kind of the state, different additional information is included. To give some examples: a numeric metric state can contain the observation time of the value; the fulfillment of the condition is indicated by an alert condition state; whether the signal presence is currently on, off, latched, or acknowledged is indicated by an alert signal state. There is also information indicating whether the corresponding component is currently on, off, paused, etc. (activation state). This allows turning functionalities on and off during runtime, like simple metrics or even whole channels, VMDs, or MDSs. Every state has in common that a state version is included. This state version is increased every time the state changes. This enables for example to determine whether information is outdated or arrived in a wrong order. Furthermore, the MDIB itself has also a version number and additionally a sequence- and instance-ID.

Specific device characteristics that would overload the generic information and service model and are only required by a small fraction of implementers can be defined as an *extension* of the model. These extensions are ignored by participants that cannot interpret them and thus offer customization options for very specific use cases. A prominent example is an extension that allows for DICOM configuration management over SDC: It adds plug-and-play capability for DICOM devices by facilitating the exchange of configuration parameters over SDC [60].

2) Service Model: According to the SOA approach, a service model has to be defined to ensure interoperability. IEEE P11073-10207 defines a get service as mandatory. This service can be used for reading access according to the request-response pattern. This means that a requesting service consumer gets a direct response from the service provider. Using this service, all specified aspects of the MDIB can be read, for example, the whole MDIB, the whole description of the device, sub-trees of the device description, single description elements. The device state and every single state element are also accessible via this service.

Additional services can be provided by the SDC devices. The event report service allows access according to the publish-subscribe pattern. In this case, the service consumer subscribes to events of the device. After the subscription, the device sends notifications to the service consumer whenever a value changes. Thus, the service consumer does not need to poll values of interest frequently. This mechanism is also used by the SOMDA device to notify subscribed network participants about alert conditions and alert signals.

The optional set service allows remote control access to the SDC device. The possible operations are defined in the Service and Control Object within the device description. Thus, only the well-defined set of remote control operations can be triggered. If a service consumer likes to invoke a remote control operation, it is required that it is subscribed to a specific operation invocation report. This mechanism ensures that the service consumer is informed about the execution state of the remote control operation. This is important because the processing time can be quite long compared to the communication time. In case of mechanical inertial systems like ventilators, pumps, OR-tables, etc. the processing can take several seconds up to minutes or even longer.

Other optional services can be used for the transmission of waveforms or context information.

C. Architecture and Binding

The standard IEEE P11073-20701 [61] describes the allover architecture and idea of the SOMDA (see Section IV-C). Furthermore, bindings are defined between IEEE P11073-10207 (DIM) and -20702 (MDPWS) (see also middle part of Fig. 3) and also to other protocols, covering aspects like time synchronization (e.g., Network Time Protocol, NTP), or Quality of Service (QoS) (e.g., Differentiated Services, DiffServ).

Specifying the bindings is necessary because there is a strict separation between IEEE P11073-10207 and -20702. The MDPWS standard IEEE 11073-20702 defines how the data is transmitted; IEEE P11073-10207 containing the Domain Information Model and the Service Model defines the structure for the exchanged data and how to get access to the data. This separation has the advantage that it is possible to exchange the transmission technology without changing the device model. As the communication stack is typically encapsulated from the (medical) application developer this ensures less effort for the development of SDC devices and service consumers if the communication technology changes. The necessary binding between IEEE P11073-10207 and -20702 is defined in IEEE P11073-20701. To give some examples what aspects are considered: The Service Model (IEEE P11073-10207) defines abstract services to get access to the SDC device. These abstract services

have to be mapped to the concrete realization of services with DPWS. Another example is the discovery process. As described in Section V-A, DPWS provides mechanisms for a dynamic discovery. To reduce network traffic, latency, and computational effort, it should be possible for service consumers to search for specific classes of medical devices. The existing DPWS mechanisms allow searching for DPWS device types. As there is a strict separation between MDPWS and the device model, a mapping of the medical device type (included in the device description) and the DPWS device type is realized.

D. Semantic Interoperability in IEEE 11073: Nomenclature and Device Specializations

Semantic interoperability can only be ensured with the help of comprehensive controlled vocabularies. Therefore, the IEEE 11073-1010X series on nomenclature has to be extended by additional term codes that allow for more types of medical devices to be described in the SDC data model. As of today, the term codes are not sufficient to fully describe surgical devices [62]. In addition to extending the IEEE 11073-1010X series, meaningful use coding systems such as the *Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT)* and *Logical Observation Identifiers Names and Codes (LOINC)* can be used in SDC.

As surgical devices are very complex, so-called *device specializations* have to be developed in addition. They specify constraints for the modeling of a specific device type: the mandatory and optional functionality, the semantics of provided metrics, the device behavior at runtime (including error handling), etc. Device specializations are defined for a class of devices, e.g., high-frequency surgical devices, endoscopic cameras, etc. As there are many manufacturers producing devices that belong to the same class, these device specializations are necessary to avoid modeling ambiguity and thus ensure the exchangeability of devices. While requiring a certain set of elements, the device specializations do not prevent a manufacturer from including other/unique additional functionality.

Currently, some device specializations are available for personal health devices in the IEEE 11073-104XX series. For more complex point-of-care medical devices, the development has been started within the IEEE 11073-103XX series. See also Table I for an overview of the IEEE 11073 standards.

VI. Safety for Dynamic Medical Device Ensembles: The OR.NET Session Concept

The patient's safety is the all-encompassing issue of medical devices. Most safety related considerations are dependent on certain risk management of the device and have to be implemented individually by the manufacturers. An important safety aspect arising in systems of dynamically and manufacturer-independent interconnected systems of medical devices is to ensure that the right devices exchange data and the right devices are affected by a remote-control command.

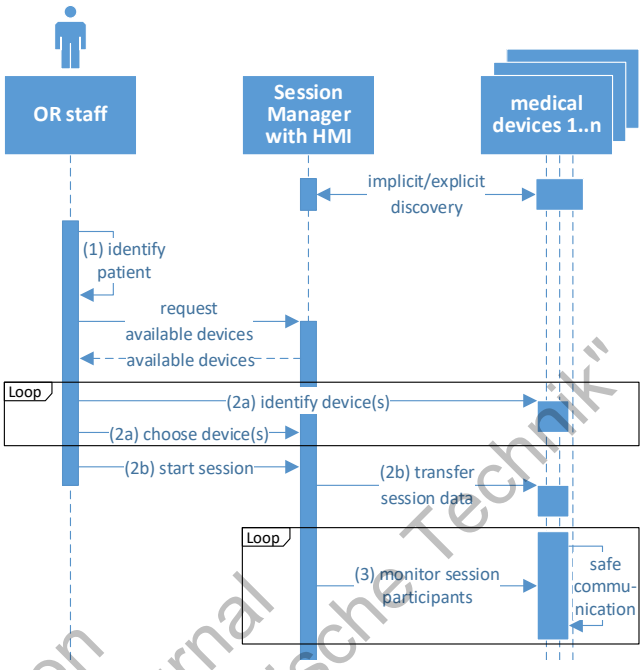

Fig. 5. *Session* creation and management process using a *Session Manager* with Human-Machine-Interface (HMI) allowing remote device selection. Steps (1) – (3) and selected additional relevant interactions are displayed. Step (4) is not included.

In this section, we describe a safety mechanism to meet this challenge.

Currently, connected medical device ensembles are strictly defined in terms of the involved devices and provided functionality. Thus, it is possible to define the device ensemble before runtime. This includes the possible and allowed remote control combinations. The SOMDA or SDC, respectively, is an enabling technology for highly dynamical interconnected manufacturer-independent medical device ensembles. The involved devices change from surgery to surgery and from patient to patient. Potentially, devices are added or removed during the surgery and can even move between ORs and patients. Thus, the assumption of predefined particular medical device ensembles does not hold anymore. Therefore, it is necessary that a distinct subset of the available medical devices (ensemble) that is used for a medical treatment can be defined dynamically. A device ensemble that is related to a patient and an order/treatment is called *Session*.

Following we will explain how a *Session* is created and managed and how it can be assured that received information belongs to the right *Session* and that only devices within the same *Session* can remote control each other.

As a precondition, we assume that there is a trusted relationship between the involved medical devices. This means that they are allowed to interconnect with each other in terms of security considerations. These security related aspects are discussed in Section VII.

A. Session Creation and Management

A *Session* is technically realized as an *EnsembleContext* defined in the IEEE P11073-10207 DIM. The responsible component for the *Session* management is the *Session Manager*. (See red network participant in Fig. 1.) In this paper we describe the concept of a *Session Manager*. Within the OR.NET project, this concept has been implemented as part of a bigger component. The workflow of creating and managing a *Session* is displayed in Fig. 5. It can be summarized as follows:

(1) Identify the patient and select an order
(2) Create the *Session*
   a) Assemble *Session* composition: identify and choose the constituent medical devices
   b) Start *Session*: transfer contextual information to the medical devices
(3) Monitor *Session* ensemble
(4) Add/remove devices to/from the *Session* if necessary

The *Session Manager* gets patient demographics data and order information from the medical IT Systems with the help of the *Demographics and Order Provider/Distributor*. For user interaction, the *Session Manager* provides a Human-Machine-Interface (HMI). The HMI can be implemented as a graphical user interface (GUI) of a PC or tablet computer, as a barcode or RFID (radio-frequency identification) reader, or any suitable realization. Using this HMI, a human actor within the OR composes the necessary medical devices for the surgical treatment. After the patient to be treated has been identified, this patient and the corresponding order is selected using the HMI in step (1). Afterwards, the necessary medical devices are chosen from the set of all available devices in step (2a). If the HMI provides remote device selection (selection is not done by activities like directly scanning the device), mechanisms for medical device identification are necessary to ensure that the correct and intended device is selected. For example, medical devices can implement a remote activate operation to trigger a state in which the device can be identified. According to the kind of medical device and its capabilities, a multitude of possible actions can be performed, caused by the activate operation command: activating a distinct LED light, displaying a textual message on a screen, etc. If the *Session* device ensemble is formed using utilities like a bar code scanner, the identification process is done implicitly. Additional convenience functionality can be realized for the session ensemble creation, like filtering for device location context, associating groups of devices to a session, etc.

The actual *Session* creation happens in step (2b): The *Session Manager* transfers the contextual information, like a *Session Key*, to the selected medical devices. In SDC-based implementations, the *Session Key* is modeled as the *identification* element of one instance of the *Ensemble Context* (defined in the IEEE P11073-10207 DIM). This particular *Ensemble Context* represents the *Session* for the device. The transfer of the *Session Key* requires that the devices implement a corresponding set operation for this context. This defines a minimum device capability to take part in an *OR.NET Session*. Only devices that fulfil this requirement shall be selectable in step (2a). While the *Session* is patient and order related, the corresponding *Ensemble Context* does not contain information about patient or order. Transmission of this information is not part of the core concept of a *Session Manager*.

Devices being part of a *Session* are typically necessary for the surgical treatment. Thus, it is potentially very safety critical if such devices become unavailable, due to connection-loss, software or hardware failure, etc. Therefore, the *Session Manager* monitors the devices of the *Session* (step (3)) and provides information and potentially alarm if the *Session* device ensemble changes unintendedly. The described monitoring functionality could also be done by an independent component. As devices can be removed or added to the *Session* willingly during the surgical treatment, it is possible to re-arrange the device ensemble using the HMI of the *Session Manager* (step (4)).

B. Session Safety Mechanisms

The safety goal of the *Session Context* is to ensure that the right devices communicate with each other. Such communication has two aspects:

(1) Clients shall receive, analyze, and display data of intended devices
(2) Remote control operations shall only be possible between *Session* participants

Both aspects have high influence on the patient's safety. If information from another OR is displayed, the physicians' decisions will be made based on this wrong information. This can lead to treatments harming the patient. If a control panel from one OR can unintendedly remote control devices in another OR, this can seriously harm patients in both ORs.

From the risk management point of view, the client that receives data from another device is responsible for the first aspect. In contrast to this, it is the responsibility of the device that receives a remote control command to decide whether this command is valid or not, including the second aspect. For both considerations the *Session Context* is a suitable instrument.

1) Receiving Data from Session Participants: It can be assumed that all relevant medical devices are grouped in the same *Session* according to the mechanisms described in Section VI-A. Before the client invokes a get service, respectively, before it subscribes to an event service to get information from the device, like vital signs, device parameters, etc., the client reads the *Session ID* stored in the corresponding ensemble context representing the *Session*. If the *Session ID* fits to the *Session* of interest, the client can be sure that the received data belongs to the right context. Typically, this means that it belongs to the right surgical treatment in the right OR with the right patient. As ensembles can change dynamically, the client has to monitor the *Session Context* of the devices from which it receives data. Therefore, the client subscribes to change events of the *Session Context*.

2) Remote Control Commands from Session Participants: To ensure that only remote control commands from *Session* participants are executed, the *Session Context* is combined with the *MDPWS SafetyContext* mechanism. The *MDPWS SafetyContext* allows the service providing device to define the requirement that a remote control command has to contain the *Session ID* in a certain field in the command message header. A received remote control command will only be executed if the right *Session ID* is included in the *SafetyContext* message header field. Otherwise, the device will reject the remote control command as the command has potentially been sent by a client not belonging to the same *Session* device ensemble. Note, to ensure that the *Session Context* information cannot be read and emulated by network participants for an unauthorized intrusion into the *Session*, an encryption mechanism is provided by MDPWS (see also Section VII).

VII. Security Considerations

This section is about security related aspects of SOMDA-based interconnection. A complete discussion of this broad topic would be far beyond the scope of this paper, but we want to give an overview of possible threats and discuss different solutions to overcome the security threats.

A. Eligible Devices

We distinguish two levels of communication: A consumer can read metric values published by a provider or a consumer can control a provider through its provided operations. Reading values can be considered quite safe for the provider in terms of risk management, because the consumer is responsible for proper usage of the data. In the latter case of controlling a device, regulatory issues require the vendors of the provider to check the eligibility of the consumer. A possible solution is this following concept of eligibility: A medical device is considered to be eligible to control another device, if the two device vendors agree on that. The eligibility is decided purely based on the vendor of a device. If we extend this eligibility restriction to reading metrics, too, we can reuse this idea to distinguish different security threats.

B. Possible Threats

In this section we present a very broad categorization of possible security threats. This categorization will then help us in the next section to discuss and classify security concepts trying to solve these issues.

Privacy. Leaking confidential information about patients.

(1) An eligible device is able to read values of the wrong device, e.g., a device in another *Session* (see Section VI) or a device in the wrong room.
(2) An external attacker is able to read out confidential information
   a) by pretending to be an authorized device or
   b) by listening to another communication.

Security. Harming the patient during an operation or treatment.

(3) An eligible device controls the wrong device, e.g., a device in another *Session*, or a device in the wrong room.
(4) An external attacker is able to control a device
   a) by pretending to be an authorized device or
   b) by replaying a recorded message.
(5) An external attacker manipulates patient records stored in the clinical IT system which are used to compute proper treatment of the patient.

Because of the lack of an intentionally malicious device the cases (1) and (3) could be seen as pure safety issues. Nevertheless, they are included in this categorization to stress the point, that checking eligibility is not enough to ensure safe and secure communication. Furthermore, considering foreign devices brought into a clinic, these devices must neither be able to control any device of the clinic, nor be able to read out any data without prior authorization, even if the foreign devices are eligible. So, every clinic must be able to decide which particular physical devices (identified by their UDI) are allowed to exchange information.

C. Security Concepts

Based on the classification above, we can now discuss different security aspects.

a) Divided Networks: A theoretical countermeasure to all threats would be a physically separated network for those devices which should be allowed to communicate. As we need wireless networks and the possibility to move devices from one location to another with ease, such a solution is inappropriate.

Does the need to have bigger devices networks automatically lead to an integration of the existing clinical IT network with the devices network? Strictly divided networks between IT and SOMDA would prevent us from bad influences from the IT network (5), but on the other hand we want to transfer patient and order data from the IT network to the devices. This leads us to the question of how much vertical integration we actually need. The current public discussion tends to move against a complete vertical integration, arguing that automation of critical infrastructure should not be accessible from IT networks. In our case, we need some kind of information flow, but we do not need any component in the clinical IT infrastructure being able to access individual medical devices.

It is sufficient if we only allow a very limited set of gateways and connectors between these two networks: HL7 and DICOM gateways which allow information flow in both directions, as well as *Demographics and Order Provider/Distributor* and *Observation Reporter* which translate HL7 to SDC and vice versa. All of these do not allow any remote control and they interpret all messages. This blocks anything invalid including most malicious commands. Syntactically correct but semantically malicious information system records still remain an open problem. The only solution to that is either to trust certain records of the information systems or to require manual checks for every imported data.

b) Encrypted Communication: Of course, encryption solves the main privacy issue of communications being overheard (2b). In the SOMDA, we implement the SOA messages with WS-* technologies, which means we can replace HTTP with HTTPS in order to use TLS encryption. Using a SOMDA could be understood as security by design, because we can fall back to existing strategies and implementations for securing a SOA.

Gaining information from listening to other communications (2b) might still be possible if an attacker analyzes only the meta data of the communication. For example, the timing patterns of set messages might already be enough to infer the current operation or treatment.

c) Certificate Based Authentication: In order to prevent external attackers from reading values from (2a) or controlling (4a) a device, certificate based authentication can be used. With X.509 certificates we establish a hierarchical chain of trust, which allows devices to securely identify other eligible devices. Vendors issue certificates to all other vendors, whom they trust on a level that allows controlling their devices. In order to check the eligibility of other devices, every device needs a build-in list of root certificate authorities. Note, the eligibility is purely decided based on a device vendor. As already mentioned in the discussion about eligible devices, this solution is driven by both regulatory requirements regarding safety issues and security concerns. See below for further discussion about that.

Replay attacks (4b) are prevented by signed messages in combination with proper encryption, too: With challenge-response protocols or timestamps one can assure that every new message must be different from all previously sent messages.

Like with encryption, there is no need to reinvent the wheel: *MDPWS* has build-in support for certificate handling based on existing SOA solutions.

d) White List Based Authorization: As already argued in the categorization of possible threats above, we need a way to grant authorization on the level of individual physical devices. A very simplistic approach is a white list of allowed consumers kept up to date in every provider. With the certificate based authentication described above, we can now trust the information provided by the consumers. So, the part of the access control management, which needs to be performed on the devices, can be as simple as checking if the device's identifier is contained in the white list. On the downside there is a serious drawback of this solution: The white lists must be kept up to date on all the individual providers. This could be simplified through a remote management access, e.g., the white list itself can be stored in a metric, which can be set by clients which are authenticated with the appropriate certificates, but the main problem persists: The list must be kept exhaustive and up to date on all devices.

Such a white list in parts solves the problem of consumers reading values from the wrong device (1) or even controlling the wrong device (3). However, the more safety-related question of how to make sure that always the right device is read out or controlled is not fully solved with white lists.

Fig. 6. OR.NET demonstrator at conhIT exhibition 2016, Berlin

Imagine mobile devices which are used in different rooms. Those devices should be generally able to communicate with devices in multiple rooms, but probably only with those being located in the same room right now. As discussed earlier, the *Session* concept can be used to solve this problem: One or more mobile devices can be manually added to an existing *Session* if they enter a room. The problem of how to force a device out of the *Session* if the device leaves the room still remains open.

e) Further Approaches: One might want to extend the usage of the certificates for identification of the individual devices and their authorization, but those approaches lead to several considerations: Assume vendor A trusts vendor B who then issues individual certificates for every physical device. This does not increase A's level of trust, because A still needs to trust B to assign the certificates correctly, so we can stick to simpler solutions, which use the device identifier for that.

Another approach worth mentioning is that the clinic operator assigns the certificates to the individual physical devices. With this approach, the clinic operator gains a lot more freedom in how to combine the devices, but we get the problem that all vendors need to trust the clinic operator, that their device will never be controlled by an ineligible device. Furthermore, the clinic operator is now responsible for all the certificate assignment including possible revocation.

All this completely ignores the authentication and authorization of the people using the devices, which is another important topic. Being able to remote control devices in the operation room raises a lot of new questions related to the internal command structure of the people working in the operation room. Here, we follow the assumption that devices can only be remote controlled by other devices being in the same physical room, but the SOMDA solution can easily be extended to telemedicine applications.

VIII. Concept Validation: Open Source Frameworks and Demonstrators

The mechanisms described in this paper have been validated in several demonstrators during the OR.NET project. The demonstrators are built up using three different open source frameworks implementing the IEEE 11073 SDC

standards. Using three different interoperable libraries shows that the new standards are well implementable.

- openSDC [63] (Java)
- OSCLib [64] (C++)
- SoftICE [65] (Java, wrapper available for C#)

Currently, the majority of medical devices are connected to the SDC network using an additional hardware component, typically a single-board computer or an embedded development board. This component has an interface to the proprietary manufacturer-dependent communication and an SDC interface. The translation is done in the application software using one of the mentioned libraries. Devices having a suitable network interface can be connected by a software update containing the SDC implementation. In the future, the manufacturers will integrate the SDC interface implementation directly in the device firmware [66].

During the OR.NET project, a continuous proof of the developed concepts has been done with local and supra-regional demonstrators. Comprehensive demonstrators with the complexity of a today's OR showing all aspects of the medical device interconnection have been built up at different places in Germany during the OR.NET project: at the conhIT exhibitions in Berlin in 2015 and 2016, at the University of Aachen, at the ICCAS in Leipzig, and at the University of Heidelberg. The demonstrator at the ICCAS is a permanent demonstrator and can be used for development, research, and teaching for the next years. Additionally, demonstrators in Munich and Lübeck focused on partial aspects. During the OR.NET project, more than 30 different medical devices from over 20 different manufacturers have been modeled and implemented as SDC network participants. The demonstrator at conhIT exhibition in 2016 showed this full complexity. It can be seen in Fig. 6. All aspects of the OR.NET project are shown: Device-to-device communication based on the IEEE 11073 SDC standards and on the real-time capable SRTB, as well as the interconnection between the medical devices and the clinical information systems. The demonstrators have been used for the technical validation. Additionally, multi-perspective qualitative evaluations have been conducted for technical and clinical aspects with different stakeholders like medical staff, technical staff, and operators [20]. A detailed description of the OR.NET demonstrators can be found at Rockstroh et al. [67].

IX. Conclusion and Future Work

In this paper, we presented the universal concept for safe and dynamic manufacturer-independent medical device interoperability that has been developed within the German flagship project OR.NET. The characteristics of the resulting Service-Oriented Medical Device Architecture (SOMDA) have been introduced as well as the IEEE 11073 SDC family of standards containing IEEE P11073-10207, -20701, and -20702. SDC is the first technical specification implementing the SOMDA paradigm, describing a completely distributed communication middleware for medical device interoperability. In addition, safety and security aspects have been discussed; we introduced the session concept to ensure safe communication within a particular ensemble of medical devices without disturbance and interference from other participants. The concepts described herein have also been successfully transferred into practice. Clinical and technical staff as well as hospital operators positively evaluated and validated the OR.NET approach. All stakeholders agreed that the results will provide significant advantages over existing systems. While OR.NET focused on medical devices within the OR, the developed concepts are designed to be used for all kinds of Point-of-Care (PoC) medical devices.

For a market adoption of the OR.NET interoperability concepts, further efforts are still needed: The standardization of the IEEE 11073 SDC family has to be finished. Currently, MDPWS (IEEE 11073-20702) is an approved draft standard, while the other two are still in the standardization process. Nomenclatures have to be extended and device specializations have to be developed to ensure semantic interoperability. In addition, methods for approval and certification procedures as well as risk management for open interconnected medical devices have to be refined, accompanied by testing concepts and environments. As open interoperable systems of medical devices are currently unavailable, a formal adoption of the new approval strategies has to take place. The final step will be reached when the manufacturers implement the new technology for currently available devices and develop new functionalities and products enabled by the interoperable architecture. This will finally improve patient safety, therapeutic outcome, and clinical workflows while lowering the cost of healthcare provision.

Acknowledgment

This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) as part of the OR.NET project. Grand numbers: 16KT1235, 16KT1236, 16KT1238, 16KT1239.

References

[1] C. K. Christian, M. L. Gustafson, E. M. Roth, T. B. Sheridan, T. K. Gandhi, K. Dwyer, M. J. Zinner, and M. M. Dierks, "A prospective study of patient safety in the operating room," *Surgery*, vol. 139, no. 2, pp. 159–173, feb 2006.

[2] Y.-Y. Hu, A. F. Arriaga, E. M. Roth, S. E. Peyre, K. A. Corso, R. S. Swanson, R. T. Osteen, P. Schmitt, A. M. Bader, M. J. Zinner, and C. C. Greenberg, "Protecting Patients from an Unsafe System: The Etiology & Recovery of Intra-Operative Deviations in Care," *Annals of surgery*, vol. 256, no. 2, pp. 203–210, aug 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3415974/

[3] A. Guédon, L. Wauben, M. Overvelde, J. Blok, M. van der Elst, J. Dankelman, and J. van den Dobbelsteen, "Safety status system for operating room devices," *Technology and Health Care*, vol. 22, no. 6, pp. 795–803, 2014.

[4] R. A. Weerakkody, N. J. Cheshire, C. Riga, R. Lear, M. S. Hamady, K. Moorthy, A. W. Darzi, C. Vincent, and C. D. Bicknell, "Surgical technology and operating-room safety failures: a systematic review of quantitative studies," *BMJ Quality & Safety*, vol. 22, no. 9, pp. 710–718, sep 2013. [Online]. Available: dx.doi.org/10.1136/bmjqs-2012-001778http://qualitysafety.bmj.com/lookup/doi/10.1136/bmjqs-2012-001778

[5] I. Wubben, J. G. van Manen, B. J. van den Akker, S. R. Vaartjes, and W. H. van Harten, "Equipment-related incidents in the operating room: an analysis of occurrence, underlying causes and consequences for the clinical process," *BMJ Quality & Safety*, vol. 19, no. 6, dec 2010. [Online]. Available: http://qualitysafety.bmj.com/lookup/doi/10.1136/qshc.2009.037515dx.doi.org/10.1136/qshc.2009.037515

[6] H. Lemke and M. Vannier, "The operating room and the need for an IT infrastructure and standards," *International Journal of Computer Assisted Radiology and Surgery*, vol. 1, no. 3, pp. 117–121, 2006. [Online]. Available: http://dx.doi.org/10.1007/s11548-006-0051-7

[7] K. Cleary, A. Kinsella, and S. K. Mun, "OR 2020 workshop report: Operating room of the future," *International Congress Series*, vol. 1281, pp. 832–838, may 2005. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0531513105005388dx.doi.org/10.1016/j.ics.2005.03.279

[8] K. Lesh, S. Weininger, J. M. Goldman, B. Wilson, and G. Himes, "Medical Device Interoperability-Assessing the Environment," in *2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2007)*. IEEE, jun 2007, pp. 3–12. [Online]. Available: http://dx.doi.org/10.1109/HCMDSS-MDPnP.2007.22http://ieeexplore.ieee.org/document/4438159/

[9] "Broschüre: OR.NET - Sichere dynamische Vernetzung in Operationssaal und Klinik," Tech. Rep., 2014. [Online]. Available: http://www.ornet.org/wp-content/uploads/2014/04/Broschre{_}final{_}2.0.pdf

[10] "Medical Device "Plug-and-Play" Interoperability Program working on "safe interoperability" to improve patient safety." [Online]. Available: http://www.mdpnp.org/

[11] J. Okamoto, K. Masamune, H. Iseki, and Y. Muragaki, "Development of a next-generation operating room "Smart Cyber Operating Theater (SCOT)" - development concept and project summay," in *Proceedings of CARS 2015: International Conference and Exhibition on Computer Assisted Radiology and Surgery*, Barcelona, 2015, pp. 156–158.

[12] M. Koeny, J. Benzko, M. Czaplik, B. Marschollek, M. Walter, R. Rossaint, K. Radermacher, and S. Leonhardt, *The Smart Operating Room: smartOR*. Boca Raton, FL, USA: CRC Press - Taylor & Francis Group, 2013, ch. Chapter 12, pp. 291–315.

[13] "DOOP-Projekt (Dienst-orientierte OP-Integration)." [Online]. Available: http://www.doop-projekt.de/

[14] A. Janß, S. Plogmann, and K. Radermacher, "Human-centered risk management for medical devices new methods and tools," *Biomedical Engineering / Biomedizinische Technik*, vol. 61, no. 2, jan 2016. [Online]. Available: http://www.degruyter.com/view/j/bmte.2016.61.issue-2/bmt-2014-0124/bmt-2014-0124.xml

[15] P. Knipp and A. Janß, "Regulatory Approval Route and Strategy for Open Networked Medical Devices," in *49. Jahrestagung der Deutschen Gesellschaft für Biomedizinische Technik (DGBMT), BMT 2015*, Lübeck, 2015.

[16] J. Dell'Anna, A. Janß, H. Clusmann, and K. Radermacher, "A Configurable Footswitch Unit for the Open Networked Neurosurgical OR Development, Evaluation and Future Perspectives," *i-com*, vol. 15, no. 3, p. 227, jan 2016. [Online]. Available: file://www.degruyter.com/view/j/icom.2016.15.issue-3/icom-2016-0031/icom-2016-0031.xmlhttp://www.degruyter.com/view/j/icom.2016.15.issue-3/icom-2016-0031/icom-2016-0031.xmldx.doi.org/10.1515/icom-2016-0031

[17] J. Benzko, L. Krause, A. Janß, B. Marschollek, P. Merz, J. Dell'Anna, and K. Radermacher, "Modular user interface design for integrated surgical workplaces," *Biomedical Engineering / Biomedizinische Technik*, vol. 61, no. 2, jan 2016. [Online]. Available: http://www.degruyter.com/view/j/bmte.2016.61.issue-2/bmt-2014-0125/bmt-2014-0125.xml

[18] J. Dell'Anna, A. Janß, E.-M. Zeissig, K. Ganser, K. Radermacher, and H. Clusmann, "Development of new concepts for safe and usable human-machine-interfaces in the open networked neurosurgical OR," in *67th Annual Meeting of the German Society of Neurosurgery (DGNC)*, Frankfurt am Main, 2016. [Online]. Available: http://dx.doi.org/10.3205/16dgnc357

[19] A. Will, R. Pahontu, and B. Bergh, "Vernetzte Medizintechnik im Krankenhaus: Vernetzung von Medizingeräten und weiteren IT-Komponenten," *KU Gesundheitsmanagement*, pp. 54–56, 2015. [Online]. Available: http://www.genios.de/fachzeitschriften/artikel/KU/20150105/vernetzte-medizintechnik-im-kranken/3135599939.html

[20] M. Rockstroh, S. Franke, M. Hofer, A. Will, M. Kasparick, B. Andersen, and T. Neumuth, "OR.NET: multi-perspective qualitative evaluation of an integrated operating room based on IEEE 11073 SDC," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 8, pp. 1461–1469, aug 2017. [Online]. Available: http://link.springer.com/10.1007/s11548-017-1589-2

[21] Healthcare Information and Management Systems Society (HIMSS), "HIMSS Dictionary of Healthcare Information Technology Terms, Acronyms and Organizations, Third Edition," p. 75, 2013.

[22] National Committee on Vital and Health Statistics (NCVHS), "Report on Uniform Data Standards for Patient Medical Record Information," pp. 21–22, 2000.

[23] iData Research, "US Market for Video, High-Tech and Integrated & Hybrid Operating Theatre Equipment," Tech. Rep., 2016. [Online]. Available: http://www.idataresearch.com/product/us-video-high-tech-and-integrated-operating-theatre-equipment-market-2016-forecasted-to-2022-medsuite/http://blog.idataresearch.com/stryker-karl-storz-lead-us-integrated-operating-room-market-dem and-increases-date-tec

[24] ASTM International (American Society for Testing and Materials), "ASTM F2761-09(2013), 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," West Conshohocken, PA. [Online]. Available: http://www.astm.org/Standards/F2761.htm

[25] OMG - Object Management Group, "Data Distribution Service (DDS) Specification." [Online]. Available: http://www.omg.org/spec/{#}DDS

[26] MD PnP Program, "OpenICE." [Online]. Available: www.openice.info

[27] B. Ibach, J. Benzko, and K. Radermacher, "OR-Integration based on SOA - Automatic detection of new Service Providers using DPWS," in *International Journal of Computer Assisted Radiology and Surgery*, ser. International Journal of Computer Assisted Radiology and Surgery, vol. 1. CARS. Geneva (Switzerland): Springer, 2010, pp. 195–196.

[28] M. Gessat, S. Bohn, A. Vorunganti, S. Franke, and O. Burgert, "TiCoLi: an open software infrastructure for device integration in the digital OR," *International Journal of Computer Assisted Radiology and Surgery*, vol. 6, no. 1, p. 284, 2011.

[29] "TeKoMed – Technologische Kompatibilität in der Medizintechnik durch serviceorientierte Architekturen." [Online]. Available: http://kosse-sh.de/projekte/tekomed/

[30] J. H. Pfeiffer, M. E. Dingler, C. Dietz, and T. C. Lueth, "Requirements and architecture design for open real-time communication in the operating room," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Zhuhai: IEEE, dec 2015, pp. 458–463. [Online]. Available: http://ieeexplore.ieee.org/document/7418810/

[31] Health Level Seven International (HL7), "HL7 FHIR." [Online]. Available: www.hl7.org/fhir/

[32] J. H. Pfeiffer, M. Kasparick, B. Strathen, C. Dietz, M. E. Dingler, T. C. Lueth, D. Timmermann, K. Radermacher, and F. Golatowski, "OR.NET RT: how service-oriented medical device architecture meets real-time communication," *Biomedical Engineering / Biomedizinische Technik*, 2017. [Online]. Available: http://www.degruyter.com/view/j/bmte.ahead-of-print/bmt-2017-0016/bmt-2017-0016.xml

[33] B. Andersen, M. Kasparick, H. Ulrich, S. Franke, J. Schlamelcher, M. Rockstroh, and J. Ingenerf, "Connecting the Clinical IT Infrastructure to a Service-Oriented Architecture of Medical Devices," *Biomedical Engineering / Biomedizinische Technik*, 2017.

[34] A. Janß, J. Thorn, M. Schmitz, A. Mildner, J. Dell'Anna-Pudlik, M. Leucker, and K. Radermacher, "Extended device profiles and testing procedures for the approval process of integrated medical devices using the IEEE 11073 communication standard," *Biomedical Engineering / Biomedizinische Technik*, jan 2017. [Online]. Available: http://www.degruyter.com/view/j/bmte.ahead-of-print/bmt-2017-0055/bmt-2017-0055.xml

[35] T. Erl, *SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2007.

[36] ——, *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.

[37] OASIS, "Reference Model for Service Oriented Architecture 1.0 - OASIS Standard," 2006. [Online]. Available: http://docs.oasis-open.org/soa-rm/v1.0/

[38] I. Melzer, *Service-orientierte Architekturen mit Web Services*, 4th ed. Heidelberg: Spektrum Akademischer Verlag, 2010. [Online]. Available: http://link.springer.com/10.1007/978-3-8274-2550-8

[39] W3C Working Group Note, "Web Services Glossary," 2004. [Online]. Available: http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/

[40] World Wide Web Consortium (W3C), "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language - W3C Recommendation," 2007. [Online]. Available: https://www.w3.org/TR/2007/REC-wsdl20-20070626/

[41] ——, "Web Services Description Language (WSDL) 1.1 - W3C Note," 2001. [Online]. Available: http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[42] OASIS, "UDDI Version 3.0.2 - OASIS Technical Specification," 2004. [Online]. Available: https://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm

[43] World Wide Web Consortium (W3C), "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) - W3C Recommendation," 2007. [Online]. Available: http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[44] S. de Deugd, R. Carroll, K. E. Kelly, B. Millett, and J. Ricker, "SODA: Service Oriented Device Architecture," *Pervasive Computing, IEEE*, vol. 5, no. 3, pp. 94–96, 2006.

[45] C. Mauro, A. Sunyaev, J. M. Leimeister, and H. Krcmar, "Standardized Device Services - A Design Pattern for Service Oriented Integration of Medical Devices," in *2010 43rd Hawaii International Conference on System Sciences*. Honolulu, HI: IEEE, jan 2010, pp. 1–10. [Online]. Available: http://ieeexplore.ieee.org/document/5428401/

[46] OASIS, "Devices Profile for Web Services Version 1.1 - OASIS Standard," 2009. [Online]. Available: http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.html

[47] ——, "Web Services Dynamic Discovery (WS-Discovery) Version 1.1," 2009. [Online]. Available: http://docs.oasis-open.org/ws-dd/discovery/1.1/os/wsdd-discovery-1.1-spec-os.html

[48] S. Schlichting and S. Pöhlsen, "An architecture for distributed systems of medical devices in high acuity environments - A Proposal for Standards Adoption," in *HL7, 2014. 11073/HL7 Standards Week*, San Antonio, Texas, USA, 2014. [Online]. Available: www.hl7.org/documentcenter/public/wg/healthcaredevices/20140116Anarchitectureforddistributedsystemsofmedicaldevicesinhigh-acuityenvironments.pdf

[49] M. Kasparick, S. Schlichting, F. Golatowski, and D. Timmermann, "New IEEE 11073 standards for interoperable, networked point-of-care Medical Devices," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Milan, Italy: IEEE, aug 2015, pp. 1721–1724. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/26736609http://ieeexplore.ieee.org/document/7318709/

[50] ——, "Medical DPWS: New IEEE 11073 Standard for Safe and Interoperable Medical Device Communication," in *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*. Tokyo, Japan: IEEE, oct 2015, pp. 212–217. [Online]. Available: http://ieeexplore.ieee.org/document/7390446/

[51] M. Kasparick and B. Andersen, "Offene Standards im vernetzten Operationssaal," *ELEKTRONIK PRAXIS*, vol. Nr. 14, pp. 62–64, 2016.

[52] World Wide Web Consortium (W3C), "Web Services Policy (WS-Policy) 1.5 - Framework - W3C Recommendation," 2007. [Online]. Available: http://www.w3.org/TR/2007/REC-ws-policy-20070904/

[53] ——, "Efficient XML Interchange (EXI) Format 1.0 (Second Edition) - W3C Recommendation," 2014. [Online]. Available: http://www.w3.org/TR/2014/REC-exi-20140211/

[54] IEEE Standards Association, "IEEE Standard 11073-20702-2016 - IEEE Approved Draft Standard for Medical Devices Communication Profile for Web Services," 2016. [Online]. Available: https://standards.ieee.org/findstds/standard/11073-20702-2016.html

[55] ——, "IEEE Project P11073-10207 - Standard for Domain Information & Service Model for Service-Oriented Point-of-Care Medical Device Communication." [Online]. Available: https://standards.ieee.org/develop/project/11073-10207.html

[56] ——, "Standard for ISO/IEEE Health Informatics - Point-of-care medical device communication - Part 10201: Domain information model," *ISO/IEEE 11073-10201:2004(E)*, pp. 1–183, 2005.

[57] ——, "ISO/IEEE Health Informatics - Point-Of-Care Medical Device Communication - Part 10101: Nomenclature," *ISO/IEEE 11073-10101:2004(E)*, pp. 1–492, 2004.

[58] 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 *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Orlando, FL: IEEE, aug 2016, pp. 2390–2394. [Online]. Available: https://doi.org/10.1109/EMBC.2016.7591211http://ieeexplore.ieee.org/document/7591211/

[59] M. Kasparick, M. Schmitz, F. Golatowski, and D. Timmermann, "Dynamic Remote Control through Service Orchestration of Point-of-Care and Surgical Devices based on IEEE 11073 SDC," in *2016 IEEE Healthcare Innovation Point-Of-Care Technologies Conference (HI-POCT)*. Cancun: IEEE, nov 2016, pp. 121–125.

[60] J. Schlamelcher, M. Onken, M. Eichelberg, and A. Hein, "Dynamic DICOM configuration in a service-oriented medical device architecture," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Milan: IEEE, aug 2015, pp. 1717–1720. [Online]. Available: http://ieeexplore.ieee.org/document/7318708/

[61] IEEE Standards Association, "IEEE Project P11073-20701 - Standard for Service-Oriented Medical Device Exchange Architecture & Protocol Binding." [Online]. Available: http://standards.ieee.org/develop/project/11073-20701.html

[62] B. Andersen, M. Kasparick, F. Golatowski, and J. Ingenerf, "Extending the IEEE 11073-1010X nomenclature for the modelling of surgical devices," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, feb 2016, pp. 244–247. [Online]. Available: http://ieeexplore.ieee.org/document/7455880/

[63] "sourceforge: OpenSDC facilitates development of distributed systems of medical devices." [Online]. Available: https://sourceforge.net/projects/opensdc/

[64] SurgiTAIX AG, "Open Surgical Communication Library (OSCLib)." [Online]. Available: https://github.com/surgitaix/osclib

[65] ——, "Software for the Integrated Clinical Environment (ICE)." [Online]. Available: https://bitbucket.org/surgitaix/softice

[66] M. Kasparick, F. Golatowski, and D. Timmermann, "Cyber-physische Systeme im OP-Saal - Ein Machbarkeitsnachweis." in *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI), Big Data - Mastering Complexity: 44th Annual Meeting of the Society for Computer Science, INFORMATICS 2014*, E. Plodereder, L. Grunske, D. Ull, and E. Schneider, Eds., vol. P-232. Stuttgart: Gesellschaft fur Informatik (GI), sep 2014, pp. 1203–1214.

[67] M. Rockstroh, S. Franke, R. Dees, A. Merzweiler, G. Schneider, M. Dingler, C. Dietz, J. Pfeifer, F. Kühn, M. Schmitz, A. Mildner, A. Janß, J. Dell'Anna-Pudlik, M. Köny, B. Andersen, B. Bergh, and T. Neumuth, "From SOMDA to application – Integration strategies in the OR.NET demonstration sites," *Biomedical Engineering / Biomedizinische Technik*, 2017.