# Medical DPWS: New IEEE 11073 Standard for safe and interoperable Medical Device Communication

Martin KasparickStefan SchlichtingFrank Golatowski and Dirk TimmermannInstitute of Applied Microelectronics<br/>and Computer EngineeringResearch Unit ofInstitute of Applied MicroelectronicsUniversity of Rostock, GermanyLübeck, GermanyUniversity of Rostock, GermanyEmail: martin.kasparick@uni-rostock.deEmail: stefan.schlichting@draeger.comEmail: firstname.lastname@uni-rostock.de

Abstract—The number of devices in an operation room (OR) and the complexity of the components and the overall system increases continuously. Today's vendor-dependent integrated ORs are expensive and not able to handle this complexity because they can only form isolated solutions. Thus a device communication for medical devices among each other and to medical information systems has to be based on open and vendor-independent standards. In this paper we will present new standards for networked Pointof-Care medical devices that will be part of the IEEE 11073 family of standards. A service-oriented device communication is defined by means of an architecture definition, a transport specification called Medical Devices Profile for Web Services (MDPWS), and a Domain Information & Service Model. The new system will make the complexity of a comprehensive OR integration manageable and thereby improve patient's safety. The focus of this paper is on MDPWS that enables a device communication for medical requirements and safety issues, like safe data transmission that will typically be used for safe remote control (dual channel and safety context), data streaming, and compact transmission. The suitability of the concept has been shown by a demonstrator with over 20 real world OR devices from more than 10 vendors.

#### I. INTRODUCTION

A today's operating room (OR) is full of medical devices from multiple vendors. The complexity and number of devices increases continuously. Currently information is only locally available at the producing device. Thus, not every actor in the OR gets the information that is needed. For example, the surgeon has no information about the patient's vital signs. A remote control of devices is hardly possible. So actors have to leave the high sterile areas to change device parameters. Furthermore the medical staff has to enter the same data (e.g. patient data) manually to several devices. This is faultprone and time wasting. Therefore, integration and interconnection between medical devices among each other and to medical information systems is indispensable. Available solutions for integrated ORs are monolithic systems with vendordependent communication protocols. But open standards and plug-and-play functionally are essential for a comprehensive OR-integration [1], [2], [3]. Vendor-independent systems will improve patient's safety, reduce costs, and market access for small and medium-sized enterprises will become easier.

The service-oriented architecture (SOA) paradigm has been proved as an enabling technology for an open and interoperable device integration. In this paper we will explain three standard proposals for the IEEE 11073 family of standards. They will enable a vendor-independent SOA-based communication for networked medical devices. IEEE 11073-10207 defines a Domain Information & Service Model, IEEE 11073-20702 specifies the data transmission technology (Medical DPWS), and IEEE 11073-20701 describes the comprehensive SOA-based architecture and the binding between the first two artifacts. This paper focuses on Medical DPWS. Furthermore we will also give a short introduction into the other two standards.

## II. STATE OF THE ART

## A. Medical Device Interoperability

Several huge research projects are working on the challenge of medical device interoperability among each other and to medical IT systems. Technology and standardization are focused. The SOA-based communication has been figured out as a suitable technology.

The multi-institutional community "Medical Device 'Plugand-Play' Interoperability Program" (MD PnP) [4] was founded in 2004 in the USA. The publish-subscribe based communication standard Data Distribution Service (DDS) [5] is used as communication technology. An open-source framework for an Integrated Clinical Environment (OpenICE) [6] is available. Currently only the first part of a planned series of at least five standards is available as ASTM standard F2761 [7].

Since 2012 the German flagship project OR.NET [8] has been working on safe and dynamic networking in the OR and hospital. Experience and know-how of pre-projects like the SOMIT [9] projects on gentile surgery with innovative technology called FUSION and orthoMIT, or the projects DOOP [10] and smartOR [11] incorporate in OR.NET. As implementation of the SOA paradigm the Devices Profile for Web Services (DPWS) [12] is used. SmartOR defined the Open Surgical Communication Bus (OSCB). The started standardization process is currently not pursued anymore. In contrast to OSCB that uses several centralized components the OR.NET architecture is designed without centralized components. The OR.NET project is one of the driving forces of the new IEEE 11073 standards. This work is partially part of OR.NET.

#### B. The IEEE 11073 Family of Standards

The ISO/IEEE 11073 family of standards "Health informatics - Point of Care (PoC) medical device communication / Personal health device communication" focuses on interoperability between medical devices and external computer

TABLE I. IEEE 11073 FAMILY OF STANDARDS (SELECTION)

IEEE Standard	Content
11073-10101	Nomenclature
11073-10201	Domain Information Model (DIM)
11073-20101	Application Profile, Communication Model
11073-30xxx	Transport Profiles
11073-10207 (new)	Domain Information & Service Model
11073-20701 (new)	Architecture & Binding
11073-20702 (new)	Medical DPWS

systems. The "core" sub-standards are summarized in Table I. IEEE 11073-10101 defines a nomenclature to enable semantic interoperable descriptions. It contains for example codes for structural device description, measurements, parameters, and units. IEEE 11073-10201 defines a basic Domain Information Model (DIM). An object-oriented tree-hierarchy-based model is specified for modeling a medical device and its information of interest, e.g. measurements, alerts, contextual information. Access to the device is defined within a service model.

Currently networked medical device systems are not considered in IEEE 11073. Communication model (IEEE 11073-20101) and transport profiles (IEEE 11073-30xxx series) are not suitable for a SOA-based communication. This is mainly addressed by the new standards IEEE 11073-20702 (MDPWS) and IEEE 11073-20701 (architecture). Furthermore the DIM is not sufficient for networked devices. Thus a new Domain Information & Service Model (IEEE 11073-10207) is defined.

#### III. SERVICE-ORIENTED COMMUNICATION FOR NETWORKED MEDICAL SYSTEMS

A SOA-based communication is suitable to fulfill the requirements for network medical devices in high acuity environments and enables interoperability and plug-and-play functionality. In this section we will give an overview of the proposed architecture that enables interoperable communication between medical devices among each other and also to the clinic information systems. The loose-coupled, non-centralized service-oriented device communication between several medical devices is shown in the left part of Fig. 1. The integration of the hospital's medical information systems infrastructure is also illustrated. For example, a patient monitor system provides vital signs and several corresponding physiological alerts. In today's system the information is only available at the producing device. With the help of the proposed system it is possible to use these data vendor-independent at multiple devices. It will be possible to display the vital signs and alerts from the patient monitor e.g. in the image of an endoscopic camera. This is a huge additional benefit for the surgeon. A remote control of device parameters is also possible (with respect to medical risk management issues), e.g. from a central OR-dashboard. The clinic information system (CIS) provides for example patient demographics data and order information accessible via an information system connector. Several medical devices need this information about the patient. The proposed interconnection will make the time-wasting and fault-prone manual input of this information unnecessary. For documentary issues it is also possible to write data back to the CIS. The access to Picture Archiving and Communication Systems (PACS) that provide image data can be done via special gateways. Re-implementing or suppressing well-established standards like DICOM is not



Fig. 1. Overview: device communication and new IEEE 11073 Standards[13]

intended. In fact DICOM can become plug-and-play capable with the use of the new system.

The proposed architecture and its mechanisms are going to be standardized as new parts of the IEEE 11073 family of standards. This is currently done in the official projects IEEE P11073-10207 (Domain Information and Service Model), P11073-20702 (Medical DPWS), and P11073-20701 (Architecture and Binding). The right part of Fig. 1 illustrates their relationship and interlocking. The focus of this paper is on the transport mechanisms suitable for medical safety requirements specified in IEEE 11073-20702 as Medical DPWS. Nevertheless to provide a general overview we will also give a brief introduction into both other standards.

## IV. DOMAIN INFORMATION & SERVICE MODEL: IEEE 11073-10207

To ensure semantic interoperability and accessibility in networked systems of medical devices the new "Standard for Domain Information & Service Model for service-oriented Point-of-Care medical device communication" (IEEE 11073-10207) has been developed. It is derived from the classical IEEE 11073-10201 DIM. In this section we give a short introduction. A detailed description can be found in [13].

Structures and elements that are necessary for the capability description and the description of the current state of a medical device are defined in this standard. The Medical Device Information Base (MDIB) describes the whole medical device by storing capability description and current state.

#### A. Medical Device Description

The capability of a medical device is described in a tree structure with the height of four. This is illustrated exemplarily in Fig. 2. The Medical Device System (MDS) as the root element can contain several Virtual Medical Devices (VMDs). Channels (Cha.) which are assigned to VMDs are physical or logical groupings of metrics (Met.). Metrics are (atomic) measurements, settings, status, or calculations of the medical device. Every element of the containment tree can be identified by a unique handle. A small example is given in Fig. 2 (in reality the tree will contain more nodes): The patient monitor MDS contains the VMDs pulse oximeter and ECG. The oxygen saturation channel groups the metrics SpO2 and perfusion index. The semantic of every element of the containment tree is described via codes that belong to coding systems, e.g. pulse rate metric: code 18442 (or MDC\_PULS\_RATE) with the unit code 2720 (or MDC\_DIM\_BEAT\_PER\_MIN) that indicates beats per minute (codes from IEEE 11073-10101).



Fig. 2. Schematic overview of device capability and state modeling (example containment tree; speech bubbles contain example content) [13]

The Service and Control Object (SCO) defines remote invocation capabilities for the whole system. It is part of the MDS node. Set operations can be defined, e.g. to set the (absolute) value of a metric or to set ranges of an alert condition. Activate operations can be used to trigger the processing of functions with arbitrary complexity on the remote device, like relative changes of metrics or triggering a white balance of a camera.

Physiological and technical alerts can be defined on the hierarchy of MDS, VMD, or Channel as alert systems. An alert system consists of alert conditions that have to be fulfilled to trigger (potentially multiple) alert signals. The alert signal defines the way the alert is calling for attention, like an audio alert signal if the pulse rate exceeds a limit.

## B. Medical Device State

The state of an element is a set of information at a given point of time. The set of all element states is called *MDState*. Every element of the containment tree has its corresponding state. The reference is done via the handle of the description element. The information that belongs to a state differs between element types. To give some common examples: a metric state includes the current observed value (e.g. pulse value: 61), its validity, and the observation time; an alert condition state indicates whether the condition currently fulfilled or not.

#### C. Service Model

The service-oriented communication model (service model) has five basic services for a remote interaction with the MDIB of a medical device. Reading access is provided by the get service and the event report service. The latter works according to the publish-subscribe pattern and is also responsible for alert notifications. Waveforms like ECG are transmitted via the waveform stream service. The set service allows writing access to manipulate parameters and invoke functions if there is a corresponding declaration in the SCO. Additionally the context service provides read and write access to context information, e.g. patient context (incl. patient demographics) or location context.

### V. MEDICAL DPWS: IEEE 11073-20702

The Medical Devices Profile for Web Services (MDPWS) is based on the OASIS Standard Devices Profile for Web Services 1.1 (DPWS) [12]. DPWS brings the SOA paradigm to embedded systems. It provides Web service messaging,

dynamic discovery of devices and services, basic functionalities for a self-description of the devices, and mechanisms to trigger and to subscribe and receive events. A minimal set of implementation constrains is defined with the focus on resource-constrained devices. The communication between networked Point-of-Care (PoC) devices has strict requirements regarding safe remote control (e.g. dual channel) and transmission of streams (e.g. ECG waveforms). On its own DPWS is not suitable for these requirements. Thus MDPWS defines extensions as well as restrictions to satisfy these requirements. For example, MDPWS adopts the security mechanisms of DPWS. But it also makes restrictions, e.g. the usage of client authentication with HTTP authentication is withdrawn in favor of using X.509.v3 certificates. The main extensions are:

- safe data transmission, typically used for safe remote control incl. single fault safe dual channel transmission and transmission of additional safety relevant contextual information (safety context)
- data streaming
- compact data transmission

These mentioned extensions of the basic transmission mechanism are optional as they will be used according to the clinical use case and derived risk management. Not every device will make use of all features but maybe of a subset. For example, a patient monitor has to provide an ECG waveform and will use the MDPWS streaming extensions however for an endoscopic light source this extension is probably useless. Thus a device has to advertise its requirements and properties so that a potential service consumer (client) can adjust itself to these requirements. In the following parts we will explain the extensions detailed by describing dynamic advertisement and the transmission mechanisms of the MDPWS extensions. For advertisement WS-Policy is used, that provides mechanisms for expressing capabilities, requirements, and general characteristics of entities in a Web services-based system. This is done by policy assertions. The policy subject species with which entity a policy can be associated, like endpoint, message, or operation. The policies will be associated with subjects for example by including them into existing metadata (WSDL).

## A. Advertising MDPWS Compliance

If a service is compliant with the MDPWS profile it has to include both *mdpws:Profile* and *dpws:Profile* assertion into its policy. Note that this requirement includes the hosting service (colloquially the device) and the hosted services (services that are provided by the device). The *mdpws:Profile* assertion has endpoint subject. The value of the *Optional* attribute is defined as "true" for this assertion. This indicates that the service supports MDPWS but it is not strictly necessary that the requesting client is also compliant with this profile.

## B. Safe Remote Control

A safe transmission of remote control commands is essential for networked medical devices. Several medical devices require a second channel of control commands. Of course this requirement endures in the case of networked systems. Other aspects are new to networked systems and are not present if a medical device is isolated and can only be controlled from its

SafetyReq		
DualChannelDef algorithm		transform
Selector 1	[	Selector n
ContextDef		
Selector 1	[	Selector n

Fig. 3. *SafetyReq* element to advertise additional safety requirements. (rectangle shape: XML element; oval: XML attribute; double bordered: mandatory)

own human interface. For example, if more than one client is able to control the height of the OR table in a relative manner it is possible that two commands sum up. This can lead to a potentially undesired situation. It can be solved when the OR table requires that the remote control command has to contain the contextual information of the height at the point of time when the client sends the remote control command. Thus the safe remote control capability of MDPWS has two aspects:

- dual channel transmission with single fault safety over one physical IP based transmission medium
- transmitting safety relevant contextual information with an remote control command (safety context)

1) Advertising Safety Requirements: A service of a device must have the possibility to announce its safety related requirements to clients. WS-Policy is used to enable this functionality. MDPWS defines a policy assertion, called SafetyReqAssertion. This assertion can be bound to message, operation or endpoint subject. SafetyReqAssertion has two optional Boolean attributes: transmitDualChannel and transmitSafetyContext. If the values of these attributes are "true", the client has to fulfill the dual channel and / or safety context requirements for a successful remote control. This basic announcement of safety requirements is included directly into the WSDL.

The detailed description of the specific safety requirement is defined in the so called *SafetyReq* element. This element will be embedded into the MDIB as an extension, e.g. into an operation description of the SCO. The content is illustrated in Fig. 3. It consists of the optional elements DualChannelDef and ContextDef. The DualChannelDef element that contains the definition of the dual channel transmission has two attributes: algorithm and transform. The algorithm that determines the value of the second channel representation is defined by a qualified name within the corresponding attribute, e.g. Base64encoded SHA-1 hash function (default algorithm). The transform attribute is a qualified name of a transformation that will be applied on the data before the algorithm is applied, e.g. exclusive XML Canonicalization (default transformation). The two-stage process with transformation and algorithm ensures that the computed value of the second channel is independent of the origin representation of the data. The selector elements (restricted XPath expression) define one or more attributes or elements inside a message for which a second channel shall be transmitted. For example, several selectors can be defined for the upper and the lower limit within a remote control command that changes the limits of an alert condition.

Analog attributes or elements can be specified to be embedded into the transmitted message as contextual information using the *selector* elements of the *ContextDef* element. For example, for a remote control operation of a pressure parameter the safety related contextual information could be the unit of



Fig. 4. *SafetyInfo* element to transmit safety relevant information. (rectangle shape: XML element; oval shape: XML attribute; double bordered: mandatory)

the parameter. By including the unit into the remote operation message it can be ensured that both communication participants use for example the unit of pressure *Pa* and not *psi*.

2) Safety Information Transmission: If a device has declared its requirement of safety information the client has to embed the required information into the messages, otherwise these messages will be replied with SOAP faults. If a corrupted message is detected in case of a dual channel transmission, the reply will also be a SOAP fault.

Safety information is transmitted with the so called *Safety-Info* element within the SOAP message header. Fig. 4 illustrates the content of a *SafetyInfo* element. Both dual channel information and contextual information can be transmitted in a one *SafetyInfo* if this is required by the medical device.

The DualChannel element can consist of one or more DCValue elements. This dual channel value element contains the value of the second channel and additionally information about how the value has been determined. The required URI attribute contains the ID of the element inside the message that this second channel information is related to. Every element that is referenced in this way will embed an ID attribute of the type xsd:ID so that it can be identified. The attributes algorithm and *transform* can be set optionally if multiple ways have been defined in the safety requirements advertisement and the default mechanisms are not used. The SafetyContext element can contain one or more CtxtValue elements, according to the specified selectors in the safety context requirements assertion, e.g. the unit or the resolution. The CtxtValue embeds the safety-relevant contextual information. CtxtValue and selector are connected to each other via the *selectorRef* attribute.

The transmission of safety information will be done within a SOAP header block of the message. If default values are used for optional attributes, these attributes may be omitted. (For further information about dual channel mechanism see [14].)

## C. Streaming

For many medical use cases it is necessary to transmit data streams, e.g. ECG or EEG waveforms. Thus MDPWS defines an announcement mechanism that enables a networked medical device to define the structure, content and transmission mechanism for streaming data to possible multiple clients. This mechanism is flexible in terms of stream management and the negotiation protocols (e.g. SOAP-over-UDP streaming, RTSP) as well as transport bindings of the stream (e.g. SOAP-over-UDP Multicast, RTP).

The stream source providing device includes the *Stream-Source* policy assertion into its WSDL as an endpoint policy subject to advertise the stream information. The *StreamSource* 

StreamDescriptions targentNamespace				
types				
xs:schema				
streamType as Global Element Declaration (GED)				
streamType 1 (streamType) (id) (element) (actionUR) StreamTransmission (type) streamAddress streamPeriod	•••• streamType n			

Fig. 5. *StreamDescriptions* element to advertise stream information. (rectangle shape: XML element; oval: XML attribute; double bordered: mandatory)

has an extension point for a *StreamDescriptions* element. This element contains the technical description of the stream source. A visualization of the *StreamDescriptions* element can be found in Fig. 5. Within this element one or more streams can be defined with the *streamType* element. The *streamType* element is identifiable by the unique *id* attribute. The mandatory attribute *streamType* declares that the stream follows the specification of the provided type (e.g. the SOAP-over-UDP specification). With this information every stream consuming client knows what kind of stream is provided by the device and how to handle it. The optional *element* attribute provides the information what content elements will be transmitted within the stream (e.g. samples of a waveform stream which could be defined in a used data model). The *actionURI* attribute gives a potential hint to the semantics implied by the stream messages.

Further specific information for the stream transmission can be defined in the optional element StreamTransmission. The contained *type* attribute references the mechanism for the stream transmission. If it is omitted the value of the *streamType* of the parent element is implied. This attribute is important if the parent element's streamType definition contains a set of possible types. If it is not omitted it must not contradict the information specified within the parent elements. For example if the *streamType* element defines SOAP-over-UDP as the used stream type, RTSP cannot be defined as stream type in the StreamTransmission element. The StreamTransmission element can contain the elements streamAddress and streamPeriod. The first specifies the address for stream transmission (e.g. soap.udp://239.239.235:12345), the latter the duration between two stream messages. It is specified as XML Schema datatype duration, e.g. PT0.02S for data published with 50 Hz.

If a stream type shall be used that is not defined in any common standard and cannot be referenced like SOAP-over-UDP it is possible to define new stream types by a Global Element Declaration (GED). For such a definition a *types* element that includes the schema definition will be included into the *StreamDescriptions*. Within the *schema* element the stream type can be defined by a GED in XML Schema.

There are several ways to transmit the stream description to the client. The MDPWS specification recommends the usage of WS-MetadataExchange mechanisms. In particular, the *StreamDescriptions* should be made available by including it into the *StreamSource* policy assertion. Either embedding the *StreamDescriptions* directly into the assertion or including a MetadataExchange reference to the data is possible. In the latter case the data can be retrieved by a HTTP GET to the specified URL. The relevant information are made available within the *Location* element that is defined in WS-MetadataExchange. The *Type* attribute has the value "wsstm:StreamDescriptions" and the reference is given by the *URI* attribute.

## D. Compact Transmission

DPWS uses a XML-based message serialization that is defined in die WS-I Basic Profile. As XML is both, humanand machine-readable, the representation is not very compact and a lot of data has to be transmitted. This can lead to several problems, e.g. high network load factor or problems with fulfilling timing requirements of events and remote control commands. Thus, MDPWS allows a compact representation of the XML information set by using the Efficient XML Interchange Format (EXI) [15]. EXI is a binary XML representation with good compression rates. The system requirements for encoding and decoding are suitable for embedded devices.

EXI uses grammar and vocabulary information from XML schema to generate automata, the so called EXI grammar. The structural information of an encoded EXI format is represented as edges between states of these automata. Structures and vocabulary that is not part of the XML schema file can be learn during processing time. New learned items are indexed when they appear the first time. Further appearances are coded by the use of the indexes. This learning mechanism makes EXI very flexible. If the automata are built up without any schema information it is called schema-less mode. If XML schema information is available for both communication participants the schema-informed mode can be used. In this case information from the XML schema file will not be included in the encoded EXI format. Thus schema-less encoding will usually lead to a lower sized representation. Schema extensions are learned through the described mechanism. MDPWS allows the usage of EXI schema-less mode with default options as well as EXI schema-informed mode with set compressed option and default values for all other options.

Both device and client can be involved in the negotiation process of a compact transmission. This is different from the negotiation of safe remote control requirements or the advertisement of a stream source because these ones are defined only on the device site of the communication. For a compact transmission this is only one aspect of the advertisement: The device can advertise the usage of the compact XML representation for a message by using a *Compression* policy assertion. The *Compression* policy assertion is defined in MDWPS and has message, operation, or endpoint policy subject. The *mdpws:Compression* element has two attributes with the following set of values:

- attribute *method*: *mdpws:EXI-sl* (EXI schema-less algorithm will be used) and *mdpws:EXI-si* (EXI schemainformed algorithm will be used)
- attribute *compression: mdpws:EXI-nocmpr* (no EXI compression will be used) and *mdpws:EXI-cmpr* (the EXI compression mode will be used)

Additionally the device includes the *Content-Encoding* HTTP-field with the value "x-exi" in the SOAP-over-HTTP header if a message contains EXI encoded content. Otherwise the client indicates its possibility to handle compact XML Infoset representation by including the value "x-exi" into the *Accept-Encoding* field of the HTTP-header of a request message. If so the device will respond to this request with content in compact representation if possible. A special case is the event subscription. If a client includes such a value



Fig. 6. Demonstrator at conhIT exhibition in April 2015, Berlin

in a WS-Eventing Subscription request, the event source will transmit events in a compact representation if possible. Thus not only the direct response is affected but also all event messages until this subscription is terminated.

### VI. ARCHITECTURE & BINDING: IEEE 11073-20701

The "Standard for Service-oriented Medical Device Exchange Architecture & Protocol Binding" (IEEE 11073-20701) defines the architecture for networked PoC medical devices and medical IT systems based on a SOA. Additionally the binding between the both other standards, the Domain Information and Service Model (IEEE 11073-10207) and the transport profile MDPWS (IEEE 11073-20702), is specified. The interlocking is illustrated in Fig. 1. Furthermore time synchronization and transport quality-of-service requirements are addressed.

#### VII. DEMONSTRATOR & REFERENCE IMPLEMENTATIONS

In the course of OR.NET project the system has been proofed by a comprehensive demonstrator at conhIT exhibition in April 2015 in Berlin. Fig. 6 shows the scenario with a complexity of a today's OR. All of the more than 20 medical devices from multiple vendors are interconnected to each other. This shows that the proposed system is suitable for complex networked medical device systems. So for each device a description has been developed incl. interaction possibilities. In the future new entries of the nomenclature are necessary for most of these devices. To give some examples use cases of the scenario (incomplete list): We display vital signs from a pulse oximeter and device parameter in the image of an endoscopic camera and in the OR-microscope image by using overlays. This concept will give more actors within the OR access to necessary information. To reduce the interactions with devices outside the highly sterile area we implemented a remote control of parameters of endoscopic and cutting devices. Additionally, we used the new protocols to realize dynamically assignable programmable control elements that can control multiple devices and uniform and comprehensive user interfaces (e.g. dashboard) that can also be used on mobile devices. The communication is based on standard Ethernet and WLAN for mobile devices. This demonstrator and also several smaller ones have been build up on two reference implementation of the new standards that are currently available: The Open System & Device Connectivity libraries (openSDC) [16] and the Open Surgical Communication Library (OSCLib) [17].

#### VIII. CONCLUSION

We presented a service-oriented device communication for networked medical PoC devices. Currently it is in the process of standardization as new parts of the IEEE 11073 family of standards. The focus of the paper is on IEEE 11073-20702 called Medical Devices Profile for Web Services (MDPWS). It enables a communication of networked medical devices that fulfills medical safety requirements. The main aspects are the safe remote control (including dual channel transmission over one physical IP based transmission medium and the transmission of safety relevant contextual information), data streaming, and compact data transmission. Furthermore the new Domain Information & Service Model (IEEE 11073-10207) and the system architecture and binding (IEEE 11073-20701) are described briefly. The proof of concept has been done in a real world demonstrator with over 20 medical devices from more than 10 vendors in the course of conhIT exhibition 2015 in Berlin (see Fig. 6). To the best of our knowledge it was the first time that vendor-independent device interoperability has been demonstrated with the complexity of a today's OR.

#### ACKNOWLEDGMENT

This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) under reference number 16KT1238 as part of the OR.NET project.

#### REFERENCES

- H. U. Lemke and M. W. 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.
- [2] R. M. Satava, "The Operating Room of the Future: Observations and Commentary," *Surgical Innovation*, vol. 10, no. 3, pp. 99–105, 2003.
- [3] D. Rattner and A. Park, "Advanced Devices for the Operating Room of the Future," *Surgical Innovation*, vol. 10, no. 2, pp. 85–89, 2003.
- [4] MD PnP Program, "Medical Device "Plug-and-Play" Interoperability Program," 2015-03-14. [Online]. Available: www.mdpnp.org/
- [5] Object Management Group, "Data Distribution Service (DDS) Specification." [Online]. Available: http://www.omg.org/spec/#DDS
- [6] MD PnP Program, "OpenICE." [Online]. Available: www.openice.info
- [7] ASTM, "ASTM F2761-09(2013), Part 1: General requirements and conceptual model," West Conshohocken, PA, 2013.
- [8] "OR.NET Sichere dynamische Vernetzung in Operationssaal und Klinik," 2015-03-23. [Online]. Available: www.ornet.org
- [9] "Schonendes Operieren mit innovativer Technik (SOMIT)." [Online]. Available: http://www.gesundheitsforschung-bmbf.de/de/1631.php
- [10] "DOOP-Projekt (Dienst-orientierte OP-Integration)," 15.03.2015.
  [Online]. Available: http://www.doop-projekt.de/
- [11] M. Koeny, J. Benzko, M. Czaplik, M. Walter, K. Radermacher, R. Rossaint, and S. Leonhardt, "Getting Anesthesia Online: The smartOR Network," *International Journal On Advances in Internet Technology*, vol. 5, no. 3 and 4, pp. 114–125, 2012.
- [12] OASIS, "Devices Profile for Web Services Version 1.1," 01.07.2009.
- [13] M. Kasparick, S. Schlichting, F. Golatowski, and D. Timmermann, "New IEEE 11073 Standards for interoperable, networked Point-of-Care Medical Devices," in *Engineering in Medicine and Biology Society* (*EMBC*), 37th Annual Int. Conference of the IEEE, Milan, Italy, 2015.
- [14] S. Pöhlsen, W. Schöch, and S. Schlichting, "A Protocol for Dual Channel Transmission in Service-Oriented Medical Device Architectures based on Web Services," in 3rd Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2011), 2011.
- [15] World Wide Web Consortium (W3C), "Efficient XML Interchange (EXI) Format 1.0 - W3C Recommendation," 10.03.2011.
- [16] "OpenSDC facilitates development of distributed systems of medical devices." [Online]. Available: https://sourceforge.net/projects/opensdc/
- [17] SurgiTAIX AG, "Open Surgical Communication Library (OSCLib)," [Online]. Available: http://www.surgitaix.com/cms/index.php/osclib